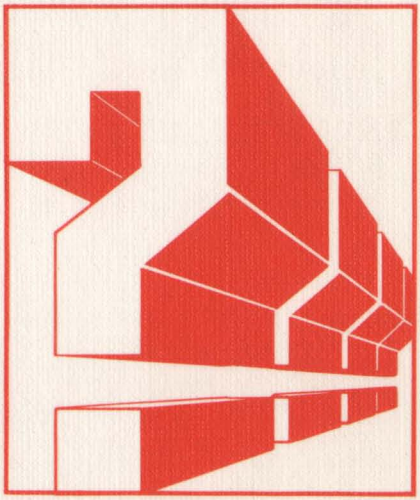
Jordan · Schellenberger

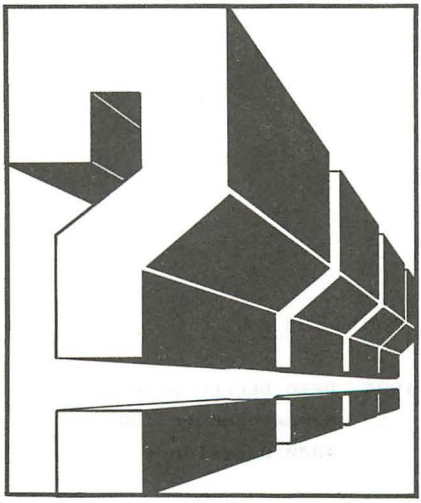
DAS DATA BECKER LEXIKON ZUM COMMODORE 64



EIN DATA BECKER BUCH

Jordan · Schellenberger

DAS DATA BECKER LEXIKON ZUM COMMODORE 64



EIN DATA BECKER BUCH

ISBN-Nr.: 3 - 89011 - 013 - 4

Copyright (C) 1984 DATA BECKER GmbH
Merowingerstr. 30
4000 Düsseldorf

Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeiner Form (Druck, Fotokopie oder einem anderen Verfahren) ohne schriftliche Genehmigung der DATA BECKER GmbH reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Wichtiger Hinweis

Die in diesem Buch wiedergegebenen Schaltungen, Verfahren und Programme werden ohne Rücksicht auf die Patentlage mitgeteilt. Sie sind ausschließlich für Amateur- und Lehrzwecke bestimmt und dürfen nicht gewerblich genutzt werden.

Alle Schaltungen, technische Angaben und Programme in diesem Buch wurden von den Autoren mit größter Sorgfalt erarbeitet bzw. zusammengestellt und unter Einschaltung wirksamer Kontrollmaßnahmen reproduziert. Trotzdem sind Fehler nicht ganz auszuschließen. DATA BECKER sieht sich deshalb gezwungen, darauf hinzuweisen, daß weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernommen werden kann. Für die Mitteilung eventueller Fehler sind die Autoren jederzeit dankbar.

V O R W O R T

"Ein richtiges Lexikon für den COMMODORE 64 müßte es geben", meinte im November letzten Jahres Thomas Jordan, der Mann unserer Werbegraphikerin, "so ein Buch, wo ganz einfach zum Nachschlagen alphabetisch geordenet alles drinsteht, was man über die Computerei im allgemeinen und den COMMODORE 64 im speziellen wissen muß, natürlich auch mit deutscher Erklärung der wichtigsten englischen Fachbegriffe". Wir fanden das eine Superidee und wußten auch gleich einen Autor: Thomas Jordan. Da er lange Zeit als Promotor in den Computerabteilungen verschiedener Kaufhäuser gearbeitet hatte, mußte er wissen, welche Fragen so ein 64er Anwender zwischen den Buchstaben A und Z hat. Als Co-Autor nahm er sich den von DATAMAT und TEXTOMAT her bestens bekannten Wolfgang Schellenberger.

Nach drei Monaten Tag und Nacht Arbeit der beiden Autoren können wird Ihnen jetzt das 350 seitige Ergebnis vorlegen, das hoffentlich auch für Sie zum nützlichen und unentbehrlichen Arbeitsinstrument wird.

Viel Erfolg bei der Arbeit mit diesem Buch und Ihrem COMMODORE 64 wünscht Ihnen

Ihr

Dr. Achim Becker

A

1.Ziffer des Hexadezimalsystems mit dem dezimalen Wert 10.
2.Taste der Tastatur des COMMODORE 64. Der *Tastaturcode dieser Taste ist 10. Außer dem Buchstaben A enthält sie folgende weitere Zeichen : a, ◻ und ♠. Diese Zeichen haben folgende *ASCII-Codes:

	Großschrift/Grapik	Groß-/Kleinschrift
A	65	193
a	-	65
◻	176	176
♠	97	-

Das recht Graphikzeichen kann durch SHIFT/A erzeugt werden, das linke durch *COMMODORE/A.
Die Zeichen, die durch diese Taste erzeugt werden können haben die folgenden *Bildschirmcodes :

	Großschrift/Grapik	Groß-/Kleinschrift
A	1	65
a	-	1
◻	112	112
♠	65	-

Die Umschaltung zwischen den beiden Modi kann durch Betätigung der Tasten *SHIFT und *COMMODORE erreicht werden.

A-D-Wandler

Ein Baustein, der ein analoges Eingangssignal in digitale Daten umsetzt. Ein analoges Eingangssignal kann zum Beispiel eine Spannung sein. Die Schwierigkeit liegt nun darin, das

unendlich fein abgestufte analoge Signal in endlich fein abgestufte digitale Daten umzuwandeln. Digitale Daten sind endlich fein abgestuft, da sie nur innerhalb betimmter Intervalle definiert sind, das heißt, es müssen Genauigkeitsschranken festgelegt werden. Ein typisches Beispiel für analog vorliegende Daten, ist die klassische Armbanduhr. oder der Tachometer

Der COMMODORE 64 verfügt nun über zwei A-D-Wandler, diese befinden sich im SID 6581. Ihre intern erzeugte Referenzspanung liegt bei etwa 2.5V. Um eine Messung mit dem COMMODORE 64 auszuwerten, ist folgende Vorgehensweise zu empfehlen :

1. Entladung der externen Kapazitäten
2. Übername des entsprechenden Wertes in Register 25 oder 26
3. Zyklische Wiederholung des Vorgehens.

Der Wert, der in die Register übernommen wird, muß der benötigten Zeit zur Wiederaufladung der externen Kapazitäten entsprechen. Näheres über den A-D-Wandler, seine Handhabung und seine Programmierung erfahren sie im 64-intern.

Abbruch

Darunter versteht man grundsätzlich das Beenden eines Programmes, dies kann auf mehrere Arten geschehen und zwar einmal durch eine definierte Abbruchbedingung, weiterhin kann es zu einem Programmabbruch durch das Auftreten einer Fehlerbedingung kommen. Eine solche Fehlerbedingung Kann ein Syntaxfehler oder ein Systemfehler sein. Drittens kann ein Programmabbruch auch extern erzeugt werden, und zwar durch Betätigung der RUN/STOP Taste. Das COMMODORE 64 meldet dann: BREAK IN XXXX.

Abbruchbedingung

Dies ist eine Bedingung innerhalb eines Programmes, die zum Programmende bzw. zu einer Programmunterbrechung führt. Eine Abruchbedingung kann zum Beispiel dann erfüllt sein, wenn eine Variabele einen vorgegebenen Wert erreicht, über- oder unterschritten hat.

Beispiel:

```
10 REM PROGRAMMSTART
20 INPUT A
30 PRINT A,SQR(A)
40 IF A<0 THEN ENDE:REM ABBRUCHBEDINGUNG
50 GOTO 20
```

In diesem Beispiel ist die Bedingung des Programmabbruchs dann erreicht, wenn der Wert der Variablen A kleiner als 0 ist, die Eingabe in unserem Beispiel also unzulässig ist.
Abbruchbedingungen können in einem Programm immer dann sinnvoll verwendet werden, um einen Endlosschleife zu beenden. Während des Programmtestes ist es ebenfalls von Nutzen das Programm an kritischen Stellen zu unterbrechen. Eine solche Unterbrechung kann durch den Befehl STOP herbeigeführt werden, der mit einer Abbruchbedingung verknüpft ist.

Abfrage

1. Test, ob eine Variabele einen bestimmten Wert angenommen hat, oder ob ein Register einen bestimmten Wert enthält.

Beispiel 1: PASSWORT

```
10 INPUT NAME$
20 IF NAME$="HANS" THEN PRINT "ALLES KLAR !"
30 GOTO 10
```

In diesem Beispiel wird die Variable NAME$ daraufhin abgefragt, ob sie die Zeichenkette "HANS" enthält.

Beispiel 2:

```
10 IF PEEK(203)= 64 THEN 10
20 PRINT "ICH BIN STARTKLAR"
30 ...
40 ...
```

Hier wurde das Register 203 daraufhin abgefragt, ob es den Wert 64 enthält.

2.Test, ob ein Peripheriegerät betriebsbereit ist.
Der COMMODORE 64 nimmt diesen Test zum Beispiel dann vor wenn das Diskettenlaufwerk angesprochen wird oder wenn der Drucker benutzt werden soll. Erhält der Computer eine Rückmeldung von dem angesprochenen Peripheriegerät, so erscheint die Fehlermeldung : DEVICE NOT PRESENT.

Abkürzung der BASIC-Befehle

Der COMMODORE 64 bietet die Möglichkeit die meisten BASIC-Befehle abzukürzen. Grundsätzlich werden die BASIC-Schlüsselworte dadurch abgekürzt, daß der erste

Buchstabe normal und der zweite Buchstabe mit *SHIFT eingetippt wird. Wie von jeder Regel gibt es auch hier Ausnahmen, die im Folgenden dargestellt werden.

BASIC-Befehl	Abkürzung
CLOSE	CL SHIFT O
GOSUB	GO SHIFT S
LEFT$	LE SHIFT F
PRINT	?
RESTORE	RE SHIFT S
RETURN	RE SHIFT T
STEP	ST SHIFT E
STR$	ST SHIFT R

Wurden die Abkürzungen innerhalb eines Programmtextes benutzt, so werden sie im Listing, egal ob auf dem Bildschirm oder dem Drucker, ausgeschrieben.

Abort

Programmabbruch

ABS

BASIC Schlüsselwort zur Ermittlung des Absolutbetrages einer Zahl bzw. eines numerischen Ausdruckes. Der Absolutbetrag gibt, mathematisch betrachtet, den Wert einer Zahl, unabhängig von ihrem Vorzeichen, wieder. Eine mögliche Anwendung des Absolutbetrages in einem BASIC-Programm, ist die Simulation des Rechnens mit *komplexen Zahlen.

SYNTAX :

ABS(Zahl) oder ABS(numerischer Ausdruck)

Beispiel:

```
10 INPUT ZAHL
20 WURZEL=SQR(ABS(ZAHL))
30 IF ZAHL<0 THEN PRINT WURZEL;"*i"
40 IF ZAHL>=0 THEN PRINT WURZEL
50 GOTO 10
```

In diesem Beispiel wurde das Rechnen mit komplexen Zahlen simuliert und zwar werden im obigen Programm auch Quadratwurzeln aus negativen Zahlen gezogen, obwohl dies normalerweise nicht möglich ist. Das Hindernis wurde dadurch umgangen, daß durch den Absolutbetrag die negativen Zahlen in quasi positive Zahlen umgewandelt wurden.

Absolute Address

*Absolute Adresse

Absolute addressing

*Absolute Adressierung

Absolute programming

Absolute Programmierung

Access

Zugriff

Access method

Zugriffsart

Access time

Zugriffszeit

ADA

Eine *höhere Programmiersprache, die erst 1980 entwickelt wurde. Sie findet Anwendung in der *Programmierung von *Großrechnern, im Bereich der *Micro- und *Homecomputer hat sie noch kaum Verbreitung gefunden.

Adder

Addierer

Add instruction

Befehl zur Addition

Adress arithmetic

Berechnung von Adressen

Adressbus

Ein Bündel von Leitungen, das den *Mikroprozessor mit den einzelnen Systemkomponenten verbindet. Der Adressbus des COMMODORE 64 ist ein 16-Bit Adressbus, das heißt, es können 2^16(65536) Adressen maximal angesprochen werden. Die Aufgabe des Adressbuses ist es, gesteuert vom *Betriebssystem und von der *CPU, die jeweils benötigten Systemkomponenten anzusprechen.

Die Systemkomponenten des COMMODORE 64 sind:

1.VIC 6569
2.SID 6581
3.COLOR-RAM 2114
4.CIA 6526
5.EXPANSION PORT
6.RAM Bereich

Address counter

*Befehlszähler

Adresse, absolute

Darunter versteht man den genau definierten Platz innerhalb eines *Registers. Dieser Platz wird mit einer Nummer bezeichnet. Das *Adressregister des COMMODORE 64 beinhaltet die Zahlen von 0 bis 65535. Der Inhalt einer Adresse kann durch *PEEK gelesen werden.

Beispiel :

```
PRINT PEEK(203)
```

Der Inhalt der Adresse 203 gibt den *Tastaturcode der momentan gedrückten Taste an. Verändert werden kann der Adressinhalt durch *POKE.

Beispiel :

```
POKE 197,28
```

Durch diesen Befehl wird in die Adresse 197 der Wert 28 geschrieben, dadurch wird die *RUN/STOP Taste blockiert.
Die Adresse wird in Programmen häufig benutzt, um bestimmten Speicherstellen bestimmte Werte zuzuordnen. Oftmals werden mit Hilfe der absoluten Adresse Funktionen der *Hardware kontrolliert.

Adressierung

Dies ist das Ansprechen bestimmter Speicherstellen, um den Inhalt der Speicherstelle zu lesen oder zu verändern. Es gibt drei Möglichkeiten der Adressierung :

1. *Direkte Adressierung
2. *Indirekte Adressierung
3. *Relative Adressierung.

Adressierung;direkte

Wird eine Speicherstelle durch einen *PEEK oder *POKE Befehl direkt mit ihrer "Nummer" angesprochen, so spricht man von direkter Adressierung. Soll zum Beispiel ein "*" in die linke obere Ecke des Bildschirms gesetzt werden, so muß die Speicherstelle 1024 angesprochen werden. Wird die Speicherstelle direkt adressiert, so lautet der Befehl wie folgt :

POKE 1024,42

Adressierung, indirekte

Bei dieser Adressierungsart bilden zwei Speicherstellen der *Zeropage den Zeiger auf die neue Adresse. Bei der indirekten Adressierung wird der Operand in Klammern gesetzt. Wird auf diese Art adressiert, so wird aus einer Speicherstelle der Zeropage ein Wert entnommen und der nächsten Speicherstelle ebenfalls. Diese Werte werden miteinander verknüpft und ergeben die Adresse der Speicherstelle, die angesprochen werden soll.

Adressierung;indizierte

Bei der indizierten Adressierung wird dem Wert, der in dem Befehl angegeben ist, der Inhalt des ebenfalls angegebenen *Zeropage-Registers hinzuaddiert. Diese Summe wird wieder als Adresse des ersten von zwei *Bytes benutzt, mit denen gearbeitet werden soll.

Beispiel :

A = PEEK(203 + X)

In diesem Beispiel wird nicht der Inhalt der Adresse 203 geladen, sondern zu dem Wert 203 wird der Wert der Variablen X hinzugezählt, der somit entstandene Wert ist die Adresse, die angesprochen werden soll.

Addressing mode

Adressierungsart, darunter wird die Art und Weise verstanden, in der addressiert wird, man unterscheidet die *direkt, die *indirekte und die *indizierte Adressierung.

Address part

Teil einer Adresse, wie er zum Beispiel bei der indirekten Adressierung verwendet wird.

Akkumulator

Eines der meist benutzten *Register des *Mikroprozessors. Es hat quasi die Funktion eines Zwischenspeichers, das heißt, die benötigten Werte werden hier abgelegt. Der Akkumulator kann direkt nur per *Maschinensprache angesprochen werden, aus einem BASIC-Programm heraus kann er nur über *POKE-Befehle genutzt werden.

Aktualisieren

Wird der Inhalt einer *Datei auf den neuesten Stand gebracht, so wurde sie aktualisiert.

Beispiel:

Werden in eine Adressdatei neue Adressen eingetragen oder werden alte Adressen verändert, so spricht man von Aktualisierung der Adressdatei.

Akustikkoppler

Mit Hilfe des Akustikkopplers ist es möglich, zwei oder mehr Computer über große Entfernung (*Datenfernübertragung) miteinander zu verbinden. Diese Verbindung wird mit Hilfe eines Telefonhörers über das Telefonnetz hergestellt. Eine sinnvolle Anwendung dieser Technik ist die Nutzung mobiler Datenstationen, die die gesammelten Daten über einen Akustikkoppler per Telefon an den Zentralrechner weitergeben.

Algebra

Lehre von der Auflösung von Gleichungen nach vorgegebenen Regeln, z. B. Punktrechnung vor Strichrechnung. Der Commodore 64 ist nicht in der Lage Gleichungen zu lösen. Der Programmierer muß die zu lösende Gleichung so weit umformen, daß die zu berechnende Variable allein links vom Gleichheitszeichen steht. Der Computer weist dann der Variablen den Wert des Ausdruckes, der rechts vom Gleichheitszeichen steht zu.

Beispiel:

Berechnung der Grundseite verschiedener rechtwinkliger Dreiecke nach Pythagoras
Grundformel: a^2 + b^2 = c^2

BASIC-Programm:

```
10 INPUT A
20 INPUT B
30 C = SQR(A^2+B^2)
40 PRINT C
50 GOTO 10
```

Algebra;Boolesche

Der wichtigste Anwendungsbereich der Booleschen Algebra ist die Schaltalgebra, diese wiederum ist die Grundlage der Computertechnik. Die Variabelen und die Funktionen der Booleschen Algebra werden binär, das heißt nur mit den Werten 0 oder 1 dargestellt. Innerhalb eines Computers gibt es nur zwei denkbare Zustände, entweder es fließt Strom, oder es fließt kein Strom; es herrscht also entweder der Zustand 1 oder der Zustand 0. Um nun die Vorgänge innerhalb eines Computers zu beschreiben ist die Boolesche Algebra sehr geeignet.
Die wichtigsten Verknüpfungen dieser Algebra sind:

1. Die *Konjunktion (Und-Verknüpfung), die durch AND dargestellt wird;
2. Die *Disjunktion (Oder-Verknüpfung), Die durch OR dargestellt wird;
3. Die *Negation (Verneinung), die durch NOT dargestellt wird.

Der Commodore 64 beherrscht alle drei Verknüpfungen. Innerhalb eines Programmes können sie sinnvoll benutzt werden zur Verknüpfung von Bedingungen.

Beispiel:

```
10 INPUT A
20 INPUT B
30 IF A<0 AND B<0 THEN 70
40 C = SQR(A) - SQR(B)
50 PRINT C
60 GOTO 10
70 PRINT "FEHLER"
80 GOTO 10
```

In diesem Beispiel sollen die Quadratwurzeln zweier Zahlen von einander abgezogen werden, dies ist aber nur mmöglich, wenn beide Zahlen positiv sind. Diese Bedingungen (A>0 und B>0) werden durch die boolesche Konjunktion mit einander verknüpft. Die eigentliche Rechnung (C=SQR(A)-SQR(B)) wird nur dann ausgeführt, wenn beide Variablen größer als Null sind.

ALGOL

Diese *Programmiersprache gehört zu den *problemorientierten, *höheren Programmiersprachen. Ihr Hauptanwendungsbbereich ist die Lösung technisch-wissenschaftlicher Probleme. Sie wurde Anfang der 60er Jahre entwickelt und wird nur auf *Großrechnern benutzt. ALGOL ist die Abkürzung für Algorithmic Language.

Algorithmus

Grundsätzlich ist dies ein Vorgang, der nach Regeln festgelegt ist und häufig wiederkehrende Gesetzmäßigkeiten aufweist. Wird ein Algorithmus in eine *Programmiersprache

übersetzt, so entsteht der Kern eines Programms.

Beispiel:

Berechnung der Quadratwurzel von 3, ohne die Funktion *SQR zu benutzen
Sinnvoller Weise wird hier mit einem Näherungsverfahren gearbeitet. Der Lösungsalgorithmus lautet:
Wähle eine Untergrenze und eine Obergrenze, wähle eine *Schrittweite, wähle eine Genauigkeitsschranke. Addiere die Schrittweite zur Untergrenze, subtrahiere die Schrittweite von der Obergrenze, quadriere die Ergebnisse und vergleiche sie mit 3, wenn die Genauigkeitsschranke erreicht ist, beende den Vorgang, sonst starte wieder bei der Addition zur Untergrenze.

Das entsprechende BASIC-Proramm sieht wie folgt aus:

```
10 INPUT "UNTERGRENZE";U
20 INPUT "OBERGRENZE";O
30 INPUT "SCHRITTWEITE";S
40 INPUT "GENAUIGKEITSSCHRANKE";G
50 E1=U^2
60 E2=O^2
70 IF 3 - E1 <= G THEN 120
80 IF E2 - 3 <= G THEN 130
90 U = U + S
100 O = O - S
110 GOTO 50
120 PRINT "Wurzel aus 3 ist : ";E1
130 PRINT "Wurzel aus 3 ist : ";E2
```

Allocate

Zuordnen

American Standard Code for Information Interchange

Abkürzung für diesen Code ist *ASCII, mittels des ASCII-Codes werden alle Zeichen in einem *Byte verschlüsselt.

Alphanumerisch

Ein Zeichensatz der alphanumerisch ist, besteht aus Buchstaben, Zahlen und Sonderzeichen. Das Gegenteil ist ein *numerischer Zeichensatz.

Analog digital converter

Englische Bezeichnung für den *A-D-Wandler

Anlauflänge

Damit ist die Länge eines *Magnetbandes gemeint, die am Aufzeichnungskopf vorbeiläuft, bis das Band die nötige Geschwindigkeit erreicht hat, damit das erste *Bit gelesen bzw. geschrieben werden kann.

Anweisung

In einer *höheren Programmiersprache ist das ein Programmschritt. Der Ausdruck "C = A + B " stellt die Anweisung dar den Inhalt des Speicherplatzes A zu dem Inhalt des Speicherplatzes B zu addieren und das Ergebnis in den

Speicherplatz C abzulegen. Die Begriffe Anweisung und Statement können synonym gebraucht werden.

Anwendungsprogrammierung

Erstellung von Software, die sich darauf beschränkt bestimmte, fest umrissenen Anwendungsprobleme zu lösen. Dies können Probleme aus dem Bereich der Wirtschaft sein, z.B. Finanzbuchhaltung, oder aus dem technisch-mathematischen Bereich, z.B. Lösung linearer Gleichungen.

Anwendungssoftware

Darunter versteht man das Ergebnis von *Anwendungsprogrammierung, also Programme bzw. Programmpakete, die für die Lösung spezieller Probleme bzw. für spezielle Anwendungen konzipiert worden sind.
Ein Beispiel für Anwendungssoftware ist DATAMAT, ein Programm, das speziell für die *Dateiverarbeitung konzipiert wurde. Ein weiteres Beispiel ist TEXTOMAT, ein *Textverarbeitungsprogramm. Das Gegenstück zur Anwendersoftware ist die *Systemsoftware.

Anzeige

Ein Gerät, oder eine Lampe bzw. Leuchtdiode, das Daten anzeigt. Langläufig wird ein Monitor bzw. ein Fernseher als Anzeige bezeichnet. Das Sichtfenster eines *Handheld-Computers ist ebenfalls eine Anzeige. Synonym zum Ausdruck Anzeige wird der Ausdruck *Display verwendet.

Anzeigeregister

Dies *Register enthält die *Daten, die auf einer *Anzeige dargestellt werden sollen. Es ist also ein Register, das zur Steuerung des *Outputs benötigt wird.

APL

Höhere Programmiersprache, die mathematisch orientiert ist. Im Gegensatz zu *ALGOL und *FORTRAN ist APL eine *Interpreter-Sprache. APL ist die Abkürzung für "A Programming Language".

Application programm

Anwenderprogramm

Arbeitsregister

In diesem *Register werden Zwischenergebnisse gespeichert. Es gestattet einen schnellen Zugriff auf die benötigten Daten.

Arbeitsspeicher

Darunter versteht man den Bereich des Speichers der für *Programme und *Daten zur Verfügung steht. Der Umfang des Arbeitsspeichers wird gewöhnlich in *Kilobyte angegeben. Der COMMODORE 64 hat einen Arbeitsspeicher von 64 Kilobyte(KB). Bei *Großrechnern wird dem einzelnen Benutzer normalerweise

nicht der gesamte Speicherbereich als Arbeitsspeicher zur Verfügung gestellt, so daß in diesem Fall der Arbeitsspeicher ein Teil des Hauptspeichers ist.

Arithmetic overflow

Wird bei einer Rechenoperation ein Wert erzeugt, der größer ist, als der größte Wert, den ein Computer verarbeiten kann, so kommt es zu einem arithmetischen Überlauf. Die größte Zahl, die der COMMODORE 64 verarbeiten kann ist 10 ^ 38.

Arithmetical instruction

Arithmetischer Befehl, dies ist ein Befehl, der eine Rechenoperation ermöglicht.

Array

In einem Array wird eine Gruppe von Variablen zusammengefaßt. Ein anderer Ausdruck für Array ist indizierte Variabele. Sind Arrays eindimensional, so werden sie auch als Vektoren bezeichnet, sind sie mehrdimensional, so nennt man sie Matrix. Weitere gängige Namen für Arrays sind Feld und Tabelle.
Soll in einem BASIC-Programm ein Array verwendet werden, so muß der benötigte Speicherplatz reserviert werden, das geschieht durch das *DIM Statement, hier wird angegeben wieviel Zeilen und Spalten ein Array enthalten soll. Das Statement DIM A(5,8) reserviert für die Variable A Speicherplatz für 5 Zeilen mit jeweils 8 Spalten. Die Variable A kann also 40 Zahlen enthalten. Jede Zahl in dieser Variablen kann direkt angesprochen werden, mit A(3,2) wird

die Zahl angesprochen, die in der 2. Spalte der 3. Zeile steht. Arrays können immer dann sinnvoll bennutzt werden, wenn Gruppen von Zahlen miteinander verrechnet werden sollen.

Beispiel:

Berechnung des Mittelwertes von Jahresumsätzen zweier Kaufhäuser über 10 Jahre.

```
10 DIM MAT(10,2)
20 DIM UMS(2)
30 FOR I = 1 TO 10
40 FOR K = 1 TO 2
50 INPUT "UMSATZ";MAT(I;K)
60 NEXT K
70 NEXT I
80 FOR I = 1 TO 10
90 FOR K = 1 TO 2
100 UMS(K) = UMS(K) + MAT(I,K)
110 NEXT K
120 NEXT I
130 DURCHUMS = UMS(1)/10
140 PRINT "DER DURCHSCHNITTSUMSATZ VON  KAUFHAUS  1  BETRAEGT
";DURCHUMS;" DM"
150 DURCHUMS = UMS(2)/10
160 PRINT "DER DURCHSCHNITTSUMSATZ VON  KAUFHAUS  2  BETRAEGT
";DURCHUMS;" DM"
```

In diesem Beispiel wurden zwei Arrays benutzt und zwar MAT, ein zweidimensionaler Array, und UMS, ein eindimensionaler Array. In den Zeilen 10 und 20 wird der Speicherplatz für die Arrays reserviert. Mit Daten werden die Arrays in den Zeilen 30-70 belegt, die eigentliche Rechnung findet in den Zeilen 80-130 und 150 statt. Die Zeilen 140 und 160 dienen dem Ausdruck der Ergebnisse.
Bisher wurden nur Arrays betrachtet, die *numerische Werte enthielten. Das oben gesagte gilt aber auch für Arrays, die *alphanumerische Werte enthalten, der einzige Unterschied ist

der *Variablenname, bei Arrays mit alphanumerischem Inhalt muß dem Variablenname das "$" Zeichen folgen. Für diese indizierten Variablen gelten die gleichen Regeln wie für normale *String-Variable.
Soll ein Array weniger als 10 Zeilen und nur eine Spalte enthalten, so braucht er nicht dimensioniert zu werden. Arrays können bis zu 255 Dimensionen enthalten, sinnvoll ist es aber, nicht über drei Dimensionen hinaus zu gehen, da man sonst leicht den Überblick verliert.

ASCII

Hinter dieser Abkürzung verbirgt sich der Amerikanische Standard-Code für Informations-Austausch. Er codiert jedes Zeichen mit 7 *Bits. Ein bestimmtes Zeichen wird durch ein *Byte, bestehend aus den 7 Bits und einem *Paritätsbit dargestellt, der ASCII-Code kann also insgesamt 128 Zeichen codieren. Dieser Code normiert die Zeichendarstellung in Computern, das heißt, daß zum Beispiel das Zeichen "A" in allen Computern den gleichen ASCII-Code hat, und zwar 65. Es werden durch den ASCII-Code alle Zeichen veschlüsselt, die ein Computer darstellen kann, also alle Groß-und Kleinbuchstaben, alle Ziffern und alle Sonderzeichen.
Die Zeichen des COMMODORE 64 haben folgende ASCII-Codes:

Großschrift/Graphik		Groß-/Kleinschrift	
32		32	
33	!	33	!
34	"	34	"
35	#	35	#
36	$	36	$
37	%	37	%
38	&	38	&
39	'	39	'
40	(	40	(
41	)	41	)
42	*	42	*
43	+	43	+
44	,	44	,
45	-	45	-
46	.	46	.
47	/	47	
48	0	48	0
49	1	49	1
50	2	50	2
51	3	51	3
52	4	52	4
53	5	53	5
54	6	54	6
55	7	55	7
56	8	56	8
57	9	57	9
58	:	58	:
59	;	59	;
60	<	60	<
61	=	61	=
62	>	62	>
63	?	63	?
64	@	64	@
65	A	65	a
66	B	66	b
67	C	67	c
68	D	68	d
69	E	69	e
70	F	70	f
71	G	71	g
72	H	72	h
73	I	73	i
74	J	74	j
75	K	75	k
76	L	76	l
77	M	77	m
78	N	78	n
79	O	79	o
80	P	80	p
81	Q	81	q
82	R	82	r
83	S	83	s
84	T	84	t
85	U	85	u

Großschrift/Graphik		Groß-/Kleinschrift	
86	V	86	v
87	W	87	w
88	X	88	x
89	Y	89	y
90	Z	90	z
91	[	91	[
92	£	92	£
93	]	93	]
94	↑	94	↑
95	←	95	←
96	—	96	—
97	♠	97	A
98	|	98	B
99	—	99	C
100	—	100	D
101	—	101	E
102	—	102	F
103	|	103	G
104	|	104	H
105	╮	105	I
106	╰	106	J
107	╯	107	K
108	╰	108	L
109	╲	109	M
110	╱	110	N
111	╭	111	O
112	╮	112	P
113	●	113	Q
114	—	114	R
115	♥	115	S
116	|	116	T
117	╱	117	U
118	╳	118	V
119	o	119	W
120	♣	120	X
121	|	121	Y
122	♦	122	Z
123	+	123	+
124	▒	124	▒
125	|	125	|
126	◢	126	▚
127	◥	127	▞
128		128	

Großschrift/Graphik		Groß-/Kleinschrift	
160		160	
161	▮	161	▮
162	▬	162	▬
163	▔	163	▔
164	_	164	_
165	|	165	|
166	▒	166	▒
167	|	167	|
168	▪	168	▪
169	◤	169	▨
170	▌	170	▌
171	├	171	├
172	▪	172	▪
173	└	173	└
174	┐	174	┐
175	▬	175	▬
176	┌	176	
177	┴	177	┴
178	┬	178	┬
179	┤	179	┤
180	▌	180	▌
181	▮	181	▮
182	▌	182	▌
183	▔	183	▔
184	▬	184	▬
185	▬	185	▬
186	┘	186	✓
187	▪	187	▪
188	▪	188	▪
189	┘	189	┘
190	▪	190	▪
191	▚	191	▚

Assembler

Maschinenorientierte *Programmiersprache, die sich hauptsächlich *mnemotechnischer Kürzel bedient. Der Assembler ist von Prozessor zu Prozessor verschieden, da jeder seiner Befehle genau dem Format des entsprechenden Maschinenbefehls entspricht. Die Programmierung in Assembler ist schwieriger, als die Programmierung in einer *höheren Programmiersprache, da die Assemblerworte abstrakter sind als die einer anderen Programmiersprache. Ein Assemblerprogramm hat aber eine erheblich kürzere Ablaufzeit, als beispielsweise ein BASIC-Programm, da hier die Übersetzung in den *Maschinencode sehr viel schneller von statten geht.
Der Mikroprozessor, der die Seele des COMMODORE 64 bildet, ist der 6510. Ein sehr wichtiges Hilfsmittel, um diesen Prozessor in Assembler zu programmieren ist PROFIMAT.

Assemblieren

Vorgang der Übersetzung eines Assemblerprogramms in den *Maschinencode. Hierbei werden die mnemotechnischen Kürzel in den maschineninternen Binärcode übertragen und die symbolischen Adressen in absolute bzw. relative Adressen übertragen.

Assoziativspeicher

Im Normalfall werden Speicherstellen über ihre *Adresse angesprochen, es wird zum Beispiel die Speicherstelle mit der Adresse 150 angesprochen, damit mit ihrem Inhalt weitergearbeitet werden kann. der Assoziativspeicher geht nun genau entgegengesetzt vor, er spricht die Speicherstellen über ihren Inhalt an, er ruft zum Beispiel die Speicherstelle mit dem Inhalt "HANS" auf um ihr als neuen Wert "OTTO"

zuzuweisen. Bisher war es noch nicht möglich einen preiswerten Assoziativspeicher zu bauen, der die gleiche Kapazität hat, wie die herkömmlichen Speicher.

ATN

BASIC-Schlüsselwort zur Ermittlung des Arcustangens einer Zahl bzw. eines Ausdruckes. Der Arcustangens ergibt den Winkel(im Bogenmaß), dessen Tangens x ist. DerArcustangens ist also die Umkehrfunktion zum Tangens. Sinnvoll benutzt werden kann der ATN dann, wenn innerhalb trigonometrischer Berechnungen ein bestimmter Winkel ausgerechnet werden soll.

SYNTAX:

ATN(Zahl) oder ATN(Ausdruck)

Beispiel:

```
10 INPUT A
20 C = SIN(A)/COS(A)
30 WINKEL = ATN(C)
40 PRINT "DER GESUCHTE WINKEL LAUTET : ";WINKEL
50 GOTO 10
```

Aufruf

Von einem Aufruf wird dann gesprochen, wenn innerhalb eines Programms eine bestimmte Routine bzw. ein bestimmtes Unterprogramm gestartet wird. Unterprogramme werden in *BASIC meist durch *GOSUB oder durch *USR(x) aufgerufen.

Aufwärmzeit

Darunter versteht man die Zeit, die ein Gerät benötigt, um seine volle Betriebsbereitschaft zu erreichen. Das Diskettenlaufwerk benötigt zum Beispiel einen gewisse Aufwärmzeit, um die Diskette auf die benötigte Umdrehungsgeschwindigkeit zu bringen. Ein weiteres Beispiel sind die alte Röhrenfernseher, die eine gewisse Aufwärmzeit benötigten, um die Röhren auf die erforderliche Betriebstemperatur zu bringen.

Aufzeichnungsdichte

Die Aufzeichnungsdichte eines *Datenträgers wird in *Bit pro Zoll angegeben. Dieses Maß gibt die Zahl der Bits an, die auf einem Zoll Spurlänge abgelegt werden können. Üblicherweise liegt die Aufzeichnungsdichte zwischen 300 und 1600 Bit pro Zoll.

Ausdruck

1. Über einen Drucker ausgegebene Informationen, die entweder einen Kopie des Bildschirminhaltes, ein Programmlisting, einen Tabelle, ein Text oder ähnliches sein können.

2. Mathematischer Term, der als Ergebnis eine Zahl hat. Ein Ausdruck bzw. Term ist zum Beispiel:

SQR(A)/A

Beim Commodore 64 ist es möglich anstatt einer *Variablen oder einer Zahl einen Ausdruck in ein BASIC-Schlüsselwort einzutragen.

Ausfallzeit

Darunter wird die Zeit verstanden, in der ein Computer, auf Grund einer Systemstörung nicht benutzbar ist.

Ausführungszeit

Die Zeit, die ein Computer benötigt, um einen *Befehl, oder ein *Programm zu bearbeiten. Der Commodore 64 bietet durch seine eingebaute Echtzeituhr die Möglichkeit die Ausführungszeit exakt zu bestimmen.

Beispiel:

```
10 TI$ = "000000"
20 FOR I = 1 TO 1000
30 A = SQR(I)
40 NEXT I
50 PRINT TI$
```

Die Ausführungszeit dieses Programms läßt sich nun sehr leicht an Hand des Strings TI$ ermitteln. Die benötigte Zeit kann folgendermaßen abgelesen werden:
Die ersten beiden Stellen geben die Stunden wieder, die nächsten beiden Stellen die Minuten und die letzten beiden Stellen schließlich die Sekunden, die zur Programmausführung benötigt wurden.

Ausgabe

1. Lesen von *Daten aus einem Speicher und Übertragung in einen anderen Speicher bzw. Verarbeitung der gelesenen Daten

durch den Computer.

2. Anzeige der Daten auf einem *Monitor bzw. Ausdruck der Daten auf einem Drucker oder einem anderen *Peripheriegerät.

Ausgabegerät

Gerät, auf dem das Ergebnis eines *Programms oder ein Programmlisting ausgegeben wird. Übliche Ausgabegeräte des Commodore 64 sind *Monitor, Fernseher, Drucker, *Plotter, *Datasette, *Diskettenlaufwerk.

Ausgabegeschwindigkeit

Die Zahl von Daten, die pro Zeiteinheit vom Computer an ein Peripheriegerät ausgegeben werden.

Ausgaberegister

*Register, in dem die Daten solange aufbewahrt werden, bis sie vom *Ausgabegerät verarbeitet werden können.

Auswahlmaske

Begriff der *Dialogprogrammierung. Unter einer Auswahlmaske wird eine *Maske verstanden, die die Möglichkeit bietet aus mehreren Alternativen eine auszuwählen. Der Gegensatz dazu ist die *Erfassungsmaske.
Die Auswahlmaske findet immer dann Anwendung, wenn der Programmierer dem Benutzer mehrere Möglichkeiten anbieten

will.

Beispiel:

Es wird eine Logarithmentafel programmiert. Damit nicht immer das gesamte Programm durchlaufen werden muß, erhält der Benutzer die Möglichkeit aus dem Angebot das für ihn passende auszusuchen. Die Auswahlmmaske in diesem Beispiel könnte das folgenden Aussehen haben:

WURZEL	(1)
POTENZ	(2)
SINUS	(3)
COSINUS	(4)
TANGENS	(5)
COTANGENS	(6)
FAKULTAET	(7)
PROGRAMMENDE	(0)

GEBEN SIE IHRE WAHL EIN

Der Benutzer hat hier also die Möglichkeit zwischen sieben verschiedenen Alternativen zu wählen. Hat er zum Beispiel die Alternative "SINUS" gewählt, so wird der Programmteil aufgerufen, der zur Berechnung des Sinus vorgesehen ist.
Eine gute Auswahlmaske sollte dem Benutzer auch immer die Möglichkeit bieten das Programm zu beenden.

AUTO

*Systembefehl, der auf einigen Computern möglich ist, er bewirkt, das die *Zeilennummerierung automatisch vorgenommen wird. Auf dem Commodore 64 ist dieser Befehl nur in Zusammenhang mit einem erweiterten BASIC, z.b. SIMONS-BASIC, möglich.

Auto restart

*Selbststart eines Programms

Auxilary routine

Hilfsprogramm, darunter versteht man auch *Dienstprogramme, wie sie zum Beispiel von der Test/Demo Diskette der COMMODORE Diskettenlaufwerken angeboten werden. Hilfsprogramme können auch selbst erstellt werden, um zum Beispiel Druckeranpassungen vorzunehmen.

Average transfer rate

Darunter wird die durchschnittliche Datenübertragungsgeschwindigkeit verstanden.

B

1. Ziffer des *Hexadezimalsystems mit dem dezimalen Wert 11.

2. Teil der Tastatur des COMMODORE 64 mit dem Tastaturcode 28. Durch diese Taste können die Zeichen B, ▚ b, ◫ und erzeugt werden. Diese Zeichen haben die folgenden *ASCII-Codes :

	Großschrift/Graphik	Groß-/Kleinschrift
B	66	194
b	-	66
▚	191	191
◫	98	-

Die Bildschirmcodes dieser Zeichen lauten :

	Großschrift/Graphik	Groß-/Kleinschrift
B	2	66
b	-	2
▚	127	127
◫	66	-

Die Umschaltung zwischen den Modi Großschrift/Graphik und Groß-/Kleinschrift kann durch die gleichzeitige Betätigung der Tasten *SHIFT und *COMMODORE erzeugt werden.

Background processing

Darunter versteht man Hintergrundverarbeitung, das heißt, daß mit dem Drucker gearbeitet werden kann und gleichzeitig weiter mit dem Computer editiert werden kann.

Background programm

Programm, das im Hintergrund abläuft, diese Möglichkeit wird nur von Personal Computern bzw. größeren Computern angeboten.

Backup

Darunter wird grundsätzlich eine Sicherheitskopie verstanden. Sicherheitskopien können von *Dateien und von *Programmen angefertigt werden. Der Sinn der Sicherheitskopie liegt darin, daß für den Fall, daß der *Datenträger, auf dem das Programm vorliegt, zerstört bzw. beschädigt wird, immer noch eine intakte Version existiert.

BAM

Abkürzung für Block-Avalibility-Map. Sie hat die Aufgabe die Blöcke einer *Diskette als frei oder belegt zu kennzeichnen. Aktualisiert wird sie nach jeder Manipulation der Blöcke, das heißt, jedesmal, wenn auf der Diskette etwas gespeichert, gelöscht usw. wurde. Eine *Fehlermeldung wird zum Beispiel dann ausgegeben, wenn das zu speichernde Programm mehr Blocke benötigt, als noch vorhanden sind. Beim Eröffnen eines *Files wird die BAM in den *DOS-Speicher übernommen, gleichzeitig mit dem Übertragungsbefehl aktualisiert und beim Schließen des Files wieder auf die Diskette geschrieben. Befehle, die Schreib- oder Löschfunktionen haben lesen die BAM, aktualisieren sie und schreiben sie wieder auf die Diskette zurück. Die BAM ist auf Spur 18 Sektor 0 niedergelegt.

Band

Damit ist in aller Regel ein Magnetband gemeint. In Bezug auf den COMMODORE 64 und andere *Homecomputer wird darunter meist eine Audiocasette verstanden. Um einen möglichst einwandfreien Betrieb zu gewährleisten, sollten nur Casetten verwendet werden, deren Spielzeit kürzer als 90 Minuten ist und außerdem sollte auf gute Qualität des Materials geachtet werden, am besten verwendet man Chromdioxid oder Ferro - Chrom Cassetten.

Bandanfangsmarke

Dünne Folie, die direkt vor das eigentliche Magnetband geklebt ist und den Beginn des aufnahmefähigen Teils kennzeichnet.

Bandendemarke

Dünne Folie, die direkt hinter das eigentliche Magnetband geklebt ist und das Ende des aufnahmefähigen Teils der Casette kennzeichnet.

Bandgeschwindigkeit

Damit ist die Geschwindigkeit gemeint, mit der das *Band am Schreib-/Lesekopf vorbeiläuft. Bei der Übertragung von Daten von einem Bandgerät (*DATASETTE) zum Computer sollte darauf geachtet werden, daß es möglichst nicht zu Schwankungen der Bandgeschwindigkeit kommt, da sonst eine fehlerfreie Datenübertragung nicht gewährleistet ist.

Bar code

Strich- oder Balkencode, dieser Code wird von einem Belegleser gelesen. Angewendet wird diese Technik immer dann, wenn große Stückzahlen erfasst werden sollen.

Base number

Basiszahl eines *Zahlensystems

BASIC

*Höhere Programmiersprache, die sehr häufig zur *Programmierung von *Homecomputern und *Personalcomputern verwendet wird. BASIC ist die Abkürzung für "Beginners All purpose Symbolic Instruction Code". Die *Programmiersprache BASIC hat den Vorteil, daß sie sehr leicht zu erlernen ist. Ihr Hauptanwendungsbereich ist die Lösung technisch - mathematischer Probleme, es können jedoch auch wirtschaftlich orientierte Programme erstellt werden. Der Nachteil von BASIC ist aber, daß in der Grundversion keine *strukturierte Programmierung möglich ist, diese läßt sich erst in einigen erweiterten BASIC - Versionen (z.B. SIMON'S BASIC) annähernd realisieren. Die bekannteste BASIC - Version ist das MIKROSOFT-BASIC, das einen relativ großen Befehlsvorrat hat. Das BASIC des COMMODORE 64 enthält nicht alle Befehle des MIKROSOFT-BASIC, sie können aber zum Teil simuliert werden.

Batch processing

*Stapelverarbeitung

Battery backup

Wird die Stromversorgung durch zusätzliche Batterien gesichert, so spricht man von einem Battery backup.

Baud

Einheit der Datenübertragungsgeschwindigkeit vom Computer zum *Peripheriegerät und umgekehrt. Ein Baud entspricht der Übertragung eines Binärzeichens pro Sekunde.

Baudrate

Die Baudrate gibt an, wieviel Binärzeichen pro Sekunde übertragen werde. Eine Baudrate von 600 bedeutet, daß 600 Binärzeichen pro Sekunde übertragen werden. Der COMMODORE 64 hat eine Baudrate von 300 Baud.

BDOS

BDOS ist die Abkürzung für "Basic Disk Operating System". Es ist ein Teil des *CP/M *Betriebssystems. Innerhalb dieses Systems hat das BDOS die Aufgabe die Arbeit mit dem Disketten- bzw. Plattenlaufwerk zu steuern. Dazu gehören folgende Teilaufgaben:

1. Steuerung der Datenübertragung von und zur Diskette bzw. Platte
2. Organisation der Daten und Dateien
3. Steuerung des Zugriffs auf Disketten- bzw. Plattendateien
4. Prüfung der Daten und der Datenspeicherung auf Korrektheit

Das Äquivalent des COMMODORE 64 zum BDOS ist das *DOS, das ähnliche Aufgaben zu erfüllen hat.

Befehlsadresse

Die Adresse, unter der ein Befehl im Hauptspeicher angesprochen werden kann. Diese Adresse des Befehls, der als nächster ausgeführt werden soll, steht im *Befehlsadressregister.

Befehlsadressregister

Bestandteil der *CPU eines Computers bzw. eines *Mikroprozessors.
Seine Aufgabe ist es anzugeben, welcher Befehl als nächster bearbeitet werden soll. Das Befehlsadressregister wird auch Befehlszähler genannt. Seine Arbeitsweise kann in sieben Schritte unterteilt werden :

1. Ausgabe des Befehlszählerinhaltes auf den *Adreßbus
2. Erhöhung des Befehlszählers um 1
3. Übernahme der Adresse vom Adressbus durch den Speicher
4. Decodierung der Adresse und Ansprechen der entsprechenden Speicherzelle
5. Ausgabe des Inhaltes der angesprochenen Speicherzelle(Befehl in *binärer Dar stellung) auf den *Datenbus
6. Übernahme des binär dargestellten Befehls durch die CPU
7. Eintragung des Befehls in das *Befehlsregister

Befehlscode

Darunter wird die Darstellung eines Maschinenbefehls in binärer Form verstanden.
Beispiel:

Der *Assemblerbefehl NOP (No Operation) wird von den meisten Systemen durch 00000000 dargestellt. Beim COMMODORE 64 wird dieser Befehl durch 11101010 dargestellt.

Befehlslänge

Sie gibt die Länge eines Befehls in Bit an, die Befehlslänge des COMMODORE 64 beträgt 8 Bits.

Befehlsregister

Diese Register beinhaltet den *Operationscode des Befehls, der als nächster von der *CPU decodiert und ausgeführt wird.

Befehlsvorrat

Die Gesamtmenge aller Befehle, die ein bestimmter Computer versteht, wird als sien Befehlsvorrat bezeichnet. Der Befehlsvorrat des COMMODORE 64 umfaßt ca. 80 Befehle für die BASIC Programmierung.

Befehlszähler

Synonymer Ausdruck für *Befehlsadressregister

Befehlszählersteuerung

Der Teil der *CPU, der das *Befehlsadressregister mit der Adresse des nächsten Befehls, der ausgeführt werden soll, versorgt. Normalerweise wird der *Befehlszähler nach dem Lesen eines Befehls um 1 hochgezählt, bei Sprungbefehlen (z.B. *GOTO) wird die Adresse des nächsten Befehls neu geladen.

Benchmark programm

Bewertungsprogramm

Benutzer

Darunter wird der verstanden, der mit einem *Programm oder einem *Computer arbeitet, aber nicht der, der das Programm schreibt. Synonym zu dem Ausdruck Benutzer wird sehr häufig der Ausdruck Anwender benutzt.

Benutzerfreundlichkeit

Von Benutzerfreundlichkeit wird im Zusammenhang mit Programmen gesprochen. Ein Programm ist dann benutzerfreundlich, wenn der Benutzer es leicht bedienen kann, das heißt, wenn es zum Beispiel *menuegesteuert ist und

zwar in der Art und Weise, daß aus dem Menue die Arbeitsweise des Programms deutlich wird. Je seltener das Programmhandbuch benutzt werden muß, um so benutzerfreundlicher ist das Programm.

Betriebsart

Darunter versteht man die verschiedenen Arten, in denen mit einem Computer gearbeitet werden kann, zum Beispiel *Dialogbetrieb oder *Stapelbetrieb.

Betriebsbereitschaft

Der Zustand eines Gerätes, der es zuläßt, mit diesem Gerät zu arbeiten. Der COMMODORE 64 ist dann betriebsbereit, wenn die rote Lampe auf seinem Gehäuse leuchtet, das Diskettenlaufwerk ist dann betriebsbereit, wenn die grüne Lampe leuchtet.

Betriebssystem

Sammlung von Systemprogrammen, die einen Computer erst einsatzfähig machen. Diese Systemprogramme dienen zur Bewältigung der unterschiedlichsten Aufgaben :

1. Steuerung der Peripheriegeräte
2. Steuerung der Massenspeicher
3. Verwaltung der Daten
4. Organisation der Speichersteuerung
5. Ausbringen von Fehlermeldungen
6. Steuerung des Datenverkehrs zwischen den einzelnen Systemkomponenten

Die bekanntesten und am meisten verbreiteten Betriebssysteme sind *CP/M und *MS-DOS. Mit Hilfe eines Zusatzgerätes ist es möglich den COMMODORE 64 CP/M tauglich zu machen.

Bibliothek

1. Sammlung von Programmen, die die unterschiedlichsten Verwendungsmöglichkeiten haben können (z.B. Spiele, Dateiverwaltung, Textverarbeitung usw.)
2. Sammlung von Programmroutinen, die in Form von *Unterprogrammen, die ohne weiteres in den Programmablauf eingegliedert werden können, vorliegen. Solche Unterprogramme können zum Beispiel mathematische Routinen sein, oder beispielsweise auch Fehlerroutinen. Die Gemeinsamkeit dieser Routinen ist, daß sie quasi programmunabhängig sind und sehr häufig benutzt werden, es ist ohne weiteres denkbar, Fehlerroutinen zu entwickeln, die immer wieder verwendbar sind, also zum Beispiel eine Routine, die verschiedene Fehlermeldungen auf eine bestimmte Bildschirmzeile bringt.

Beispiel:

```
10 REM WURZELBERECHNUNG
20 INPUT "ZAHL";Z
30 IF Z<0 THEN 70
40 ERG = SQR(Z)
50 PRINT "DIE WURZEL VON ";Z;" IST ";ERG
60 GOTO 10
70 GOSUB 500
80 GOTO 10
500 REM***********************************
510 REM FEHLERMELDUNG
520 REM***********************************
530 Z = PEEK(214)
540 S = PEEK(211)
550 POKE 214,22
560 POKE 211,10
570 SYS 58640
580 PRINT "ZAHL UNZULÄSSIG"
590 POKE 214,Z-1
600 POKE 211,S
610 SYS 58640
620 RETURN
```

In diesem Beispiel wird ein Unterprogramm zur Fehlerbehandlung dargestellt, das so oder in ähnlicher Form praktisch universell einsetzbar ist, es muß nur von Fall zu Fall der Text der Fehlermeldung geändert werden.
In den Zeilen 530 und 540 wird die alte Cursorposition festgehalten, damit der Cursor später wieder dort placiert werden kann. In den Zeilen 550, 560, 570 wird die Ausgabe der Fehlermeldung in der Zeile 22, beginnend in Spalte 10, vorbereitet. Der Text der Fehlermeldung wird in Zeile 580 auf den Bildschirm gebracht. Die Zeilen 590, 600 und 610 bringen den Cursor wieder auf seine alte Position.

Bidirektional

Dies heißt nichts anderes, als in beide Richtungen. Ein bidirektionaler Drucker schreibt sowohl von links nach rechts, als auch von rechts nach links. Bidirektionale Datenkanäle übertragen Daten in beide Richtungen, es gibt also keinen festen Empfänger und keinen festen Sender. Ein bidirektionaler Bus läßt ebenfalls eine Übertragung in beide Richtungen zu.

Bildschirm

Der COMMODORE64 kann an jeden handelsüblichen Bildschirm angeschlossen werden. Der Bildschirm ist ein Peripheriegerät des Computers und dient in der Regel als Ausgabegerät, oder auch zur Eingabeunterstützung.

Bildschirmadresse

Adresse eines ansprechbaren Punktes auf dem Bildschirm. Der COMMODORE 64 teilt den Bildschirm in 25 Zeilen zu je 40 Spalten auf, daraus ergeben sich 1000 Bildpunkte, die direkt adressierbar sind. Es ist durch *POKE-Befehle möglich Zeichen auf den Bildschirm zubringen und zwar an jede beliebige Stelle.
Ein POKE-BEFEHL, der ein Zeichen auf eine bestimmte Stelle des Bildschirms setzen soll, hat folgendes Aussehen :

POKE A,B

Hierbei stellt A die Adresse der Bildschirmstelle, die gefüllt werden soll, dar. Die Werte der Adressen liegen zwischen 1024 und 2023, wobei 1024 das Zeichen links oben repräsentiert und 2023 das Zeichen links unten. Der Eckpunkt

rechts oben hat die Adresse 1063 und der links unten 1984. Das B in dem POKE-BEFEHL ist der *ASCII-Code des Zeichens, das dargestellt werden soll. Das folgende Programm füllt den Bildschirm von links oben nach rechts unten zeilenweise mit "*".

```
10 PRINT CHR$(147)
20 FOR I = 1 TO 1000
30 POKE 1023+I,42
40 NEXT I
```

Bildschirmauflösung

Fähigkeit den Bildschirm in Einzelpunkte zu zerlegen. Ein normaler Fernseher hat eine Bildschirmauflösung von ca. 250.000 Punkten, währen ein normaler *Monitor eine Auflösung von etwa 500.000 Bildpunkten hat, daher ist das Bild eines Monitors schärfer, als das eines Fernsehers.

Bildschirmspeicher

Der Speicher, in dem die *Bildschirmadressen hinterlegt sind.

Binär

Alle Informationen, egal ob Programm oder Daten, werden von einem Computer intern binär dargestellt. Binäre Darstellung heißt, es werden nur zwei Symbole benötigt und zwar 0 und 1. Die Zahl 5 wird binär wie folgt dargestellt : 101. Alle Zahlen werden im *Dualsystem ausgedrückt.
Beispiel :

Dezimal	Dual
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Die binäre Darstellung hat ihre Ursache darin, daß ein Computer nur zwei Zustände verarbeiten kann, entweder es fließt Strom (1) oder es fließt kein Strom (0).

Binärkomma

Der Punkt zwischen den Positionen 2^0 und 2^-1 entspricht dem Komma bei den Dezimalzahlen. Es ist also durchaus möglich auch mit Binärzzahlen mit Nachkommastellen zu rechnen.

Binary

Binär

Binary number

Binärzahl

Binary point

*Binärkomma

Binary to decimal conversion

Umwandlung einer Binärzahl in eine Dezimalzahl

BIOS

Teil des *CP/M-Betriebssystems, der die Verwendung verschiedener Peripheriegeräte möglich macht, daher ist dieser Teil auch sehr stark von der entsprechenden *Hardware abhängig.

BIT

Kleinste mögliche Informationseinheit, aus der alle anderen Informationen aufgebaut werden können. BIT ist die Akürzung für "binary digit". Alle Buchstaben können zum Beispiel durch einen Verknüpfung von acht Bits (1 *Byte) dargestellt werden. Ein BIT kann nur die Werte "0" und "1"annehmen.

Bitrate

Maß für die Datenübertragungsgeschwindigkeit, gemessen wird in Bits pro Zeiteinheit.

Blank

Leerzeichen

Blanking

Werden Zeichen unterdrückt, also anstelle der Zeichen *Blanks ausgegeben, so spricht man von Blanking

Blank line

Bei einer Leerzeile handelt es sich um eine Blank line.

Block

Im allgemeinen werden Daten nicht einzeln sondern blockweise übertragen. Dies gilt besonders für die *Datenfernübertragung. Die übertragenen Blocks beinhalten neben den Daten auch noch Test- und Steuerinformationen, zum Beispiel Informationen über den Blockanfang und das Blockende. Außerdem enthält der Block noch Informationen, an hand derer festgestellt werden kann, ob die Daten korrekt übertragen worden sind.

BLOCK-ALLOCATE

Befehl, um den *Direktzugriff auf die *Diskette nutzen zu können. Die Aufgabe des Befehls "BLOCK-ALLOCATE" ist es,

einen Block in der *BAM als belegt zu kennzeichnen, damit er nicht von einer anderen *Datei überschrieben wird. Dies ist immer dann nötig, wenn Blöcke, die nicht Teil einer Datei sind, im Direktzugriff auf die Diskette geschrieben werden und deshalb nicht automatisch als belegt gekennzeichnet sind. Die Befehle BLOCK-ALLOCATE und *BLOCK-FREE haben nur Auswirkungen auf das Überschreiben von Blocks durch das *DOS, die als belegt gekennzeichneten Blöcke können aber durch den Befehl *BLOCK-WRITE überschrieben werden.

SYNTAX :

B-A Drive Track Sektor

Drive gibt die Nummer des Laufwerks an, auf dem die anzusprechende Diskette liegt, in der Regel wird es "0" sein.
Track gibt die Nummer des Tracks an , der benötigt wird. Die Tracknummern dürfen zwischen 1 und 35 liegen.
Sektor gibt die Nummer des benötigten Sectors an, sie darf zwischen 0 und 17-21 liegen.

Beispiel:

```
100 OPEN 1,8,15
110 INPUT "TRACK, SEKTOR ";T,S
120 PRINT #1, "B-A 0";T;S
130 INPUT #1, A$,B$,C$,D$
140 PRINT A$;",";B$;",";C$;",";D$
```

In diesem Beispiel wird der Track und der Sektor eingegeben, anschließend wird der Block als belegt gekenzeichnet und abschließend wird die Meldung des Diskettenlaufwerkes gelesen. War der Block noch frei, so erscheint die Meldung "00,ok,00,00", war der Block hingegen schon belegt, so erscheint die Meldung "65,NO BLOCK,TT,SS". Hierbei gibt TT die Tracknummer und SS die Sektornummer an.

BLOCK-EXECUTE

Befehl, um den *Direktzugriff auf die Diskette nutzen zu können. Dieser Befehl dient dazu, einen Block von der Diskette in den *Puffer zu laden und ihn dort als *Maschinenprogramm auszuführen.

SYNTAX :

B-E Kanlnummer Drive Track Sektor

Bei diesem Befehl ist zu beachten, das beim Öffnen des Direktzugriffskanals die Puffernummer mit angegeben wird, wenn das *Maschinenprogramm nicht verschiebar und für einen bestimmten Puffer geschrieben worden ist.

Beispiel :

```
100 OPEN 1,8,15
110 OPEN 2,8,2, "3"
120 PRINT #1, "B-E 2 0 17 12"
```

Zunächst wird hier der Puffer 3 dem Kanal 2 zugeordnet, anschließend wird der Inhalt von Track 17, Sektor 12 in den Puffer 3 geladen und dort als Maschinenprogramm ausgeführt.

BLOCK-FREE

Mit Hilfe dieses Befehls, kann direkt auf die Diskette zugegriffen werden. Er bewirkt das Gegenteil des Befehls *BLOCK-ALLOCATE, ein Block der vorher als belegt gekennzeichnet war, wird nun wieder freigegeben.

SYNTAX :

B-F Drive Track Sector

Die Syntax ist also analog zum *BLOCK-ALLOCATE Befehl.

BLOCK-READ

Dieser Befehl wird für den *Direktzugriff auf die Diskette benötigt. Seine Aufgabe ist es einen Block von der Diskette in den *Puffer einer Direktzugriffsdatei zu lesen. Diese Direktzugriffsdatei muß natürlich vorher geöffnet worden sein. Dieser Befehl hat die Abkürzung "B-R". Der einzige Nachteil dieses Befehls ist aber, daß mit ihm das erste *Byte des Blocks nicht gelesen wird, daher ist es ratsam hier den Befehl "U1" zu benutzen.

SYNTAX :

U1 Kanalnummer Drive Track Sektor

Hierbei ist zu beachten, daß die Kanalnummer angegeben wird, die beim Öffnen der Direktzugriffsdatei verwendet wurde. Beim Diskettenlaufwerk 1541 wird für Drive immer 0 angegeben.

Beispiel :

```
10 OPEN 1,8,15
20 OPEN 2,8,2,"#"
30 PRINT#1,"U1 2 0 18 0"
40 GET#2, A$,B$
50 PRINT ASC(A$),ASC(B$)
```

Mit diesem Programm wurde der Inhalt des Track 18 Sector 0 in den Puffer gelesen, anschließend wurden durch GET#2 die ersten beiden Bytes aus dem Puffer gelesen.

Block length

Blocklänge

BLOCK-WRITE

Befehl, der den Direktzugriff auf die Diskette ermöglicht. Der BLOCK-WRITE Befehl bietet die Möglichkeit den Inhalt des *Puffers auf einen beliebigen Block der Diskette zu schreiben. Sinnvoll angewendet werden kann dieser Befehl im Zusammenhang mit dem *BLOCK-READ Befehl, es ist dann möglich mit BLOCK-READ Blöcke von der Diskette zu lesen, einige Bytes zu verändern und die Blöcke dann mittels BLOCK-WRITE wieder auf die Diskette zurückzuschreiben. Der Nachteil des BLOCK-WRITE Befehls (Abkürzung B-W) ist, daß hier in das erste Byte des Blocks der augenblickliche Inhalt des Pufferzeigers geschrieben wird, daher wird meist der Befehl U2 benutzt.

SYNTAX :

U2 Kanalnummer Drive Track Sektor

Die Syntax ist analog zum BLOCK-READ Befehl. Auch hier muß als Kanalnummer die Nummer der Direkzugriffsdatei angegeben werden.

Beispiel :

```
10 OPEN 1,8,15
20 OPEN 2,8,2,"#"
30 PRINT#2, "TESTWORT"
40 PRINT#1, "U2 2 0 1 0"
50 CLOSE 2
60 Close 1
```

Branch

*Verzweigung innerhalb eines Programms

Breakpoint

Die Stelle an der ein Programm unterbrochen wird, dies kann durch einen Befehl innerhalb des Programms geschehen, durch *STOP, aber auch durch Betätigung der RUN/STOP Taste. Der COMMODORE 64 meldet dann "BREAK IN XXX". Wurden keine Veränderungen am Programm vorgenommen, so kann es an der gleichen Stelle durch *CONTINUE wieder gestartet werden.

BTX

Abkürzung für Bildschirmtext. Bildschirmtext ist ein neues Medium, das es ermöglicht von zu Hause aus mit anderen *Computern, innerhalb bestimmter Grenzen, Kontakt aufzunehmen. Es soll zum Beispiel möglich sein den eigenen Kontostand von zu Hause aus abzufragen.

Bubble memory

Magnetblasenspeicher, Speicher, der nicht mit *Disketten oder *Bändern arbeitet, sondern mit Magnetblasen. Dieser Speicher konnte sich in der Praxis nich durchsetzen, da er nicht sicher genug ist.

Bürocomputer

Darunter versteht man *Computer mit einer *Speicherkapazität von mindestens 32 *Kilobyte. Gegenüber dem *Homecomputer besitzen sie meist schon in der Grundausstattung einen *Monitor. Ihr Einsatzgebiet ist hauptsächlich kaufmänischer Natur, also Faktura, Finanzbuchhaltung, Kostenrechnung usw.

Buffer

Puffer

BUFFER-POINTER

Dieser Befehl ist ein Hilfsbefehl zum *Direktzugriff auf die *Diskette. Er wird benötigt um den Zeiger auf eine beliebige Position im *Puffer zu setzen.

SYNTAX :

B-P Kanalnummer Position

Häufig wird dieser Befehl im Zusammemhang mit dem *BLOCK-READ Befehl oder dem *BLOCK-WRITE Befehl benutzt.

Beispiel :

```
10 OPEN 1,8,15
20 OPEN 2,8,2,"#"
30 PRINT#1, "U1 2 0 18 0"
40 PRINT#2, "B-P 2 144"
50 FOR I = 1 TO 16
60 GET#2, A$
70 IF A$ = CHR$(160) THEN 60
80 PRINT A$
90 NEXT I
100 CLOSE 2
110 CLOSE 1
```

In diesem Beispiel wurde der Name der Diskette eingelesen, dieser Name befindet sich immer in Track 18 Sektor 0 einer Diskette. Grundsätzlich ist noch festzuhalten, daß der Pufferzeiger völlig frei beweglich ist.

Bug

Wörtlich übersetzt heißt dies "Käfer", darunter wird ein Fehler im Programm verstanden. Wird ein Programm von Fehlern befreit, so nennt man diesen Vorgang Debugging.

Bus

Darunter versteht man ein Leitungssystem, das Informationen zwischen dem Computer und seiner Peripherie hin- und herbewegt, bzw. den Informationsaustausch zwischen den Systemkomponenten regelt.

Bussines data processing

Darunter wird die *kommerzielle Datenverarbeitung verstanden.

Byte

Ein Byte besteht aus acht *Bit, es stellt die kleinste Einheit dar, mit der ein *Computer gewöhnlich arbeitet. Es lassen sich zu Beispiel alle *ASCII-Zeichen in einem Byte codieren.

C

1. Ziffer des Hexadezimalsystems mit dem dezimalen Wert 12.

2. Teil der Tastatur des COMMODORE 64. Diese Taste hat den *Tastaturcode 20. Mit Hilfe dieser Taste können die Zeichen C, c, ◘ und — . dargestellt werden.Diese Zeichen haben die folgenden *ASCII-CODES :

	Großschrift/Graphik	Groß-/Kleinschrift
C	67	195
c	-	67
◘	188	188
—	99	-

Die *Bildschirmcodes der Zeichen lauten :

	Großschrift/Graphik	Groß-/Kleinschrift
C	3	67
c	-	3
◘	124	124
—	64	-

Die Umschaltung zwischen den beiden Modi kann durch gleichzeitige Betätigung der taste *SHIFT und *COMMODORE bewirkt werden.

Calculator

Rechenmaschine bzw. Taschenrechner

CALL

Befehl des MIKROSOFT-BASIC. Seine Aufgabe ist es ein Maschinenunterprogramm zu starten.

SYNTAX :

CALL Variablenname(Argumentliste)

Mit "Variablenname" wird eine numerische *Variable bezeichnet, die die Anfangsadresse des Maschinenunterprogramms enthält. Es dürfen hier keine *Arrays angegeben werden.
Die "Argumentliste" enthält die Namen der Variablen, deren Werte an das Maschinenunterprogramm übergeben werden sollen.
Das BASIC des COMMODORE 64 enthält diesen Befehl nicht, aber dafür den Befehl *USR, der eine ähnliche Aufgabe hat, jedoch von den Möglichkeiten her von geringerem Umfang ist als dieser MIKROSOFT-Befehl.

Carriage Return

Übersetzt heißt das "Wagenrücklauf". Bei einigen *Computern wird die Taste, mit der die Informationen an den Computer geschickt werden mit "Carriage Return" bezeichnet. Diese Taste hat beim COMMODORE 64 die Bezeichnung *"RETURN". Die Betätigung dieser Taste veranlaßt den Computer die eingegebenen Daten in seinen Hauptspeicher zu übernehmen und zu verarbeiten.

Carry

Übertragung von Daten

Cartridge

Bezeichnung für ein *Steckmodul.

Cassette

Massenspeicher für Daten und Programme. Die Casette wird als Massenspeicher haupsächlich im Zusammenhang mit *Homecomputern benutzt. Als Speichermedium zeichnet sie sich dadurch aus, daß sie preiswert ist. Der Nachteil der Cassette ist allerdings, ihre relativ hohe *Zugriffszeit und die Tatsache, daß auf sie nur sequentiell zugegriffen werden kann. Bei der Auswahl von Cassetten sollte darauf geachtet werden, daß sie möglichst eine Spielzeit von weniger als 60 Minuten haben. Bei höheren Spielzeiten wird das Band zu dünn, so daß es zu ärgerlichen Effekten kommen kann. Das Gerät, mit dem Cassetten für den COMMODORE 64 als Massenspeicher nutzbar gemacht werden können, ist die DATASETTE.

CBM

Bezeichnung der COMMODORE *Bürocomputer, zum Beispiel CBM 8032-SK, sowie der COMMODORE Peripheriegeräte.

CCP

Teil des *CP/M *Betriebssystems, das den *Dialog zwischen *Computer und *Benutzer steuert. Dieser Teil des CP/M Betriebssystems interpretiert die eingegebenen Systembefehle und führt sie aus. CCP ist die Abkürzung für "Console

Command Processor".

CDBL

Befehl des *Mikrosoft-BASIC, der die Aufgabe hat Argumente, die einen beliebigen numerischen Typ haben dürfen, in einen Wert vom Typ doppelte Genauigkeit umzuwandeln.

SYNTAX :

CDBL (Argument)

Als Argument wird hier der Name der *Variablen eingetragen, deren Typ verändert werden soll. Dieser Befehl existiert nicht im *BASIC des COMMODORE 64, er kann auch nicht simuliert werden, da der COMMODORE 64 nicht mit doppelt genauen Zahlen arbeitet.

Centralized data processing

Zentralisierte Datenverarbeitung, diese Form der Datenverarbeitung liegt dann vor, wenn die EDV von einem Zentralrechner gesteuert wird.

CHAIN

Befehl des *Mikrosoft-BASIC, der aus einem Programm heraus ein anderes Programm aufruft und in den *Arbeitsspeicher lädt. Mit diesem Befehl können Variable an das neu geladenen Programm übergeben werden, weiterhin ist es möglich nur Teile des alten Programms zu überlagern.

SYNTAX :

CHAIN MERGE Dateiname, Zeilennummer, ALL, DELETE erste Zeilennummer-letzte Zeilenmummer

Die MERGE Option ist wahlweise, sie bewirkt, daß alle Dateien geöffnet bleiben, aber alle Zeilen des alten Programms von den Zeilen des neuen Programms überschrieben werden, die die gleiche Zeilennummer haben.
Der Dateiname muß immer angegeben werden, da er das Programm bezeichnet, das geladen werden soll.
Alle folgenden Angaben sind wahlweise. Die Angabe Zeilenmummer kann die Nummer der Zeile enthalte, von der ab das neue Programm gestartet werden soll. Mit ALL wird angegeben, daß alle Variable vom neuen Programm übernommen werden sollen. Die DELETE-ANWEISUNG gibt an, welche Zeilen des alten Programms gelöscht werden sollen.
Dieser Befehl des *Mikrosoft-Basic steht im COMMODORE-BASIC in dieser Form nicht zur Verfügung, er läßt sich aber, mit etwas Aufwand simulieren. Für die Simulation dieses Befehle wird hier auf "Das große Floppy - Buch" verwiesen.

Character set

Zeichenvorrat eines Computers, der Zeichenvorrat des COMMODORE 64 besteht aus den Ziffern, Buchstaben und Sonderzeichen.

Check

Übersetzt heißt das Test. Tests könne für verschiedene Anwendungen durchgeführt werden, einmal um Programme zu testen oder zum zweiten um Hardware zu testen.

CHECK DISK

Dienstprogramm der Test/Demo Diskette. Dieses Programm testet jeden Block der Diskette, indem es ihn beschreibt und dann wieder liest. Der Block, der gerade bearbeitet wird und die Anzahl der bereits bearbeiteten Blöcke, wird am Bildschirm angezeigt. Es ist zu beachten, daß mit dem Programm "CHECK DISK" keine Disketten bearbeitet werden sollte, auf denen sich Programme befinden, da alle Programme überschrieben werden.

Check out

Darunter versteht man das Austesten eines Programms

Chip

Bezeichnung für integrierte Schaltkreise (IC). Die Chips sind Bausteine, durch deren Entwicklung die moderne Mikroelektronik erst ermöglicht wurde. Sie sind heute aus der Elektronik nicht mehr wegzudenken. Die integrierten Schaltkreise führen vielfältige Aufgaben des Kontrollierens, Messens und Regelns aus. Das Herzstück eines Chips sind die, aus Silizium bestehenden, Baugruppen, die dem Chip seine bestimmte Funktion verleihen. Ein Beispiel für einen Chip, der sich innerhalb des COMMODORE 64 befindet ist der 6581 (Sound Chip). Er behinhaltet alle Funktionen, die es ermöglichen, den COMMODORE 64 als Synthesizer zu benutzen. Sein Innenleben bedeckt eine Fläche von ca. 5 * 5 mm und ist

über 40 Pins mit dem Rest des COMMODORE 64 verbunden z.B dem *Adressbus, dem Datenbus und diversen Steuerleitungen.

CHR$

Funktion, die es in allen BASIC-Versionen gibt. Diese Funktion hat als Ergebnis das Zeichen, dessen *ASCII-Code angegeben wurde.

SYNTAX :

CHR$(Zahl oder Ausdruck)

Beispiel :

```
10 FOR I = 1 TO 250
20 PRINT CHR$(42),
30 NEXT I
```

Durch das Beispielprogramm werden 250 "*" hintereinander auf den Bildschirm gesetzt.

CINT

Befehl des MIKROSOFT-BASIC, seine Aufgabe ist es Variablen beliebigen numerischen Typs in Integer-Variable umzuwandeln.

SYNTAX :

CINT (Variable)

Der COMMODORE 64 kennt diesen Befehl nicht, er läßt sich aber annähernd simulieren.
Soll in einem Programm eine Variable für bestimmte Zwecke als

*INTEGER-Variable benutzt werden und für andere Zwecke als *REAL-Variable, so wäre es im MIKROSOFT-BASIC möglich diese Variable zweimal umzuwandeln, beim COMMDORE 64 hingegen muß in diesem Fall mit zwei Variable gearbeitet werden.

Beispiel :

```
10 INPUT X
20 X% = X
30 FOR I = 1 TO X%
40 PRINT I, I*0.86
50 NEXT I
60 PRINT X
```

Clear

Löschen von Zeilen, Zeichen, Bildschirm oder Disketteninhalten.

Clock

Takt, Schwingung

CLOSE

Befehl, der in allen BASIC-Versionen existiert, er hat die Aufgabe eine Datei, die nicht mehr benutzt wird, zu schließen.

SYNTAX :

CLOSE Filenummer

Als Filenummer wird die Nummer der Datei angegeben. Diese Nummer darf zwischen 1 und 255 liegen.

CLR

Befehl, der in allen BASIC-Dialekten vorkommt, er setzt alle numerischen Variablen auf Null zurück und alle *Stringvariable auf den Leerstring.
SYNTAX :

CLR

Dieser Befehl wird nach dem *RUN Statement automatisch ausgeführt, er kann aber auch in ein Programm eingebaut werden.

Beispiel :

```
10 CLR
20 INPUT A
30 INPUT B
40 C = A + B
50 PRINT C
60 GOTO 10
```

Hier werden nach jedem Rechenvorgang alle Variablen wieder auf Null gesetzt.

CLR/HOME

Taste der Tastatur des COMMODORE 64, sie setzt den Cursor in die linke obere Ecke des Bildschirms, wird diese Taste zusammen mit der *SHIFT Taste betätigt, so wird der gesamte

Bildschirminhalt gelöscht. Diese Taste hat keinen *Tastaturcode, sie kann also nicht im Programm abgefragt werden. Der ASCII-Code dieser Taste ist 19.

CMD

Schlüsselwort des COMMODORE-BASIC, es leitet Ausgaben, die eigentlich über den Bildschirm gehen sollen auf ein anderes Ausgabegerät um, das kann zum Beispiel ein Drucker sein oder die DATASETTE oder das Diskettenlaufwerk.

SYNTAX :

CMD Filenummer oder numerischer Variable

Es ist zu beachten, daß das Gerät, das nach dem CMD-Befehl angesprochen werden soll, zuvor geöffnet worden ist. Die numerische Variable muß sich auf die Filenummer beziehen.

Beispiel :

```
10 PRINT CHR$(147)
20 OPEN 1,4
30 INPUT "WURZELN DER ZAHLEN VON 1 BIS ";Z
40 CMD 4
50 FOR I = 1 TO Z
60 PRINT I,SQR(I)
70 NEXT I
80 PRINT#1
90 CLOSE 1
100 END
```

In diesem Beispiel soll eine beliebige Anzahl von Zahlen und ihre Quadratwurzeln ausgegeben werden. In Zeile 20 wird zunächst der Drucker geöffnet, in Zeile 40 wird die Ausgabe auf den Drucker umgeleitet, so daß das PRINT-STATEMENT in

Zeile 60 keine Auswirkungen auf die Bildschirmausgabe hat. In den Zeilen 90 und 100 wird der Bildschirm als Ausgabegerät wieder geschlossen, so daß alle Ausgaben wieder über den Bildschirm laufen.
Das BASIC-Schlüsselwort CMD kann auch ausserhalb eines Programms verwendet werden und zwar zum Beispiel, wenn das Programmlisting über den Drucker ausgegeben werden soll.

Beispiel :

```
OPEN 4,4
CMD 4
LIST
```

Soll nachher wieder der Bildschirm als Ausgabegerät benutzt werden, so muß der Drucker geschlossen werden und zwar durch folgende Befehlsfolge:

```
PRINT#4
CLOSE 4
```

Jetzt steht der Bildschirm wieder als Ausgabegerät zur Verfügung.

COBOL

*Höhere Programmiersprache, die hauptsächlich für kommerzielle Anwendungen entwickelt wurde. COBOL ist die Abkürzung für "COMMON BUSINESS ORIENTED LANGUAGE". Diese Programmiersprache zeichnet sich vor allem durch eine sehr komfortabele Dateiverwaltung aus. Weiterhin ist festzuhalten, daß COBOL eine *Compilersprache ist, anfangs konnte COBOL nur auf *Großrechnern verwendet werden, mittlerweile gibt es aber auch schon COBOL-Compiler für *Personalcomputer.

Code

Darunter versteht man eine Vorschrift zur Verschlüsselung von Informationen, zum Beispiel die Verschlüsselung von Zahlen in *binäre Zeichen oder die Verschlüsselung von Buchstaben in Zahlen nach dem *ASCII-Code. An jeden Code wird die Forderung der Eindeutigkeit gestellt, das heißt, es darf jedem Zeichen, das verschlüsselt werden soll, nur ein Codezeichen zugeordnet werden und umgekehrt.

Code converter

Gerät oder Programm, daß einen Code in einen anderen Code umsetzt.

Codieren

Umsetzung von Zeichen in einen *Code. Wird ein *Algorithmus in ein Programm umgesetzt, so ist das ebenfalls eine Codierung.

COMAL

*Höhere Programmiersprache, die auf *BASIC aufbaut, aber einige Erweiterungen gegenüber dieser Programmiersprache aufweist. COMAL ist die Abkürzung von "COMMON ALGORITHMIC LANGUAGE". Eine wichtige Eigenschaft von COMAL ist, daß sie die *strukturierte Programmierung zuläßt.

Command-Ebene

Der COMMODORE 64 kann auf zwei Ebenen betrieben werden, auf der *Programmebene und auf der Command Ebene. In der Command Ebene ist es nicht möglich Programme auszuführen, hier ist aber die Möglichkeit gegeben Programme zu erstellen oder Programme auszulisten. In dieser Ebene Kann der COMMODORE 64 auch als Taschenrechner benutzt werden, da er in der Lage ist einzelne Befehle auszuführen.

Comment

Kommentar

Comment Field

Feld in dem ein Kommentar steht. Im *BASIC des COMMODORE 64 werden Kommentare dadurch gekennzeichnet, daß Kommentarzeilen durch das Wort *REM begonnen werden.

COMMODORE

Taste der COMMODORE 64 Tastatur. Diese Taste hat keinen Tastaturcode. Sie hat im Zusammenhang mit der *SHIFT Taste die *ASCII-Codes 8 und 9. Ein Bildschirmcode wird ihr nicht zugeordnet.

COMMON

Anweisung des MIKROSOFT-BASIC, die nur in Zusammenhang mit dem *CHAIN-Befehl sinnvoll gebraucht werden kann. Diese Anweisung macht es möglich gezielt bestimmte Variable an ein neues Programm zu übergeben.

SYNTAX :

COMMON Variablenliste

Die Variablenliste beinhaltet die Namen der Variablen, die übergeben werden sollen. Zwischen den Variablennamen muß ein Komma stehen. Es ist darauf zu achten, daß ein bestimmter Variablenname immer nur in einer COMMON-Anweisung auftauchen darf. Es können keine einzelnen Elemente eines *Arrays übergeben werden, wohl aber der ganze Array.

Comparator Check

Prüfung durch Vergleich

Compiler

*Höhere Programmiersprachen können von einem *Computer nicht ohne weiteres verstanden werden, sie benötigen einen Compiler oder einen *Interpreter. Ein Compiler hat die Aufgabe den, in einer höheren Programmiersprache geschriebenen, Programmtext in die Maschinensprache zu übersetzen. Der Compiler übersetzt den gesamten Programmtext, überprüft ihn auf Syntaxfehler und führt anschließend das Programm aus, wenn keinen Syntaxfehler mehr im Programmtext vorhanden sind. Ist das Programm einmal fehlerfrei durch einen Compiler gelaufen, so braucht es nicht mehr neu übersetzt zu werden, wenn es später noch einmal

benötigt wird. Der Vorgang des Übersetzens wird compilieren genannt.

Compiler level languages

*Höhere Programmiersprachen, die einen *Compiler benötigen, damit sie von dem Computer verarbeitet werden können.

Computer

Ein Computer ist ein programmgesteuerter, elektronischer *Digitalrechner. Grundsätzlich besteht jeder Computer aus drei Komponenten, der Eingabeeinheit, der Zentraleinheit und der Ausgabeeinheit. Diese drei Komponenten finden sich auch beim COMMODORE 64 wieder, die Tastatur entspricht der Eingabeeinheit, das "Innenleben" der Konsole der Zentraleinheit und der Bildschirm schließlich der Ausgabeeinheit. Die *DATASETTE und das *Diskettenlaufwerk können sowohl Eingabe- als auch Ausgabeeinheit sein. Ein Drucker wäre eine mögliche weitere Ausgabeeinheit.

Conditional Jump

*Bedingter Sprung

CONT

Kommando des COMMODORE 64, es ermöglicht die Programmfortsetzung nach einem *Abbruch des Programms.

SYNTAX :

CONT

Dieses Kommando setzt das Programm an der Stelle fort, an der es unterbrochen wurde. Sind nach der Programmunterbrechung Zeilen geändert oder hinzugefügt worden, so ist das Kommando CONT nicht mehr benutzbar, das Programm muß dann wieder neu gestartet werden.

Control chracter

Steuerzeichen, diese Zeichen werden benötigt, wenn zum Beispiel Drucker angesprochen werden soll, dann wird über die Steuerzeichen beispielsweise der Seitenvorschub gesteuert.

Controlport

Der COMMODORE 64 besitzt zwei Controlports, die zu den Verbindungen des COMMODORe 64 mit der Außenwelt zählen. Diese Ports dienen dazu *Joysticks und *Paddles anzuschließen. Die Controlports sind 5-Bit-Schnittstellen, hier wird jeder Richtung des Joysticks ein Bit zugeordnet, das letzte Bit wird dem Feuerknopf zugeordnet. Außerdem besitzen die Controlports noch zwei Analogeingänge, die zur Abfrage der Paddles dienen. Der Nachteil der Ports ist, daß sie mit der Tastatur verknüpft sind, so daß die programmtechnische Handhabung nur unter Blockade der Tastatur erfolgen kann.

Control register

Befehlsregister

COPY

Befehl des COMMODORE-BASIC, der das Kopieren von *Dateien innerhalb einer *Diskette zuläßt. Mit Hilfe dieses Befehls ist es auch möglich aus mehreren sequentiellen *Files einen neuen zu bilden.

SYNTAX :

C:NEUDAT=ALTDAT1,ALTDAT2,...

Zu beachten ist hierbei, daß der Name der neuen Datei noch nicht auf der Diskette vorhanden sein darf, außerdem ist es mit diesem Befehl nicht möglich mehrere Programme zu einem zusammenzufassen.

COPY/ALL

Dienstprogramm der COMMODORE Diskettenlaufwerke, es ermöglicht das Kopieren ganzer Disketten bzw. ausgewählter Programme von einem Laufwerk auf ein anderes. Es ist zu beachten, daß die Geräteadresse eines der Laufwerke auf einen anderen Wert als 8 eingestellt werden muß. Das Programm ist sehr *benutzerfreundlich aufgebaut und ist daher praktisch selbsterklärend.

COS

BASIC-Schlüsselwort, das der Ermittlung des Cosinus eines Winkels dient. Diese Funktion wird hauptsächlich für trigonometrische und geometrische Aufgaben benötigt.

SYNTAX :

COS (Zahl oder numerischer Ausdruck)

Soll der Cosinus eines Winkels berechnet werden, der im Gradmaß dargestellt ist, so muß der Wert diese Winkels mit PI/180 bzw *ATN(1)/45 multipliziert werden. Der Grund dafür ist, daß der COMMODORE 64, wie auch fast alle anderen *Homecomputer, die Winkel im Bogenmaß darstellt. Die übliche Darstellung ist aber die im Gradmaß.

Beispiel :

```
10 PRINT CHR$(147)
20 INPUT "WINKEL";W
30 ERGEBNIS = COS(W*(ATN(1)/45))
40 PRINT "DER COSINUS VON ";W;" BETRAEGT : ";ERGEBNIS
50 GOTO 20
```

Das Beispielprogramm dient der Berechnung der Cosinuswerte verschiedener Winkel, die im Gradmaß dargestellt sind.

COSECANS

Mathematische Funktion, die im allgemeinen nicht explizit im BASIC enthalten ist. Der Cosecans gehört zu den trigonometrischen Funktionen und beschreibt das Verhältnis von Ankathete zur Hypothenuse im rechtwinkeligen Dreieck. Diese Funktion läßt sich durch die bekannten trigonometrischen Funktionen darstellen und zwar wie folgt:

COSECANS(X) = 1/SIN(X)

Beim COMMODORE 64 kann diese Funktion leicht in den Programmablauf eingebaut werden und zwar mit Hilfe des *DEF

FN-Statements.

Beispiel :

```
10 DEF FNCS(X) = 1/SIN(X)
20 INPUT "WINKEL";W
30 ERGEBNIS = FNCS(W*(ATN(1)/45))
40 PRINT "DER COSECANS VON ";W;"IST : ";ERGEBNIS
50 GOTO 20
```

Es ist darauf Acht zu geben, daß ein Computer gewöhnlich die Winkel im Bogenmaß darstellt und nicht, wie von der Logarithmentafel gewohnt, im Gradmaß. Soll sich das Ergebnis auf Gradmaßwinkel beziehen, so müssen diese mit dem Faktor ATN(1)/45 multipliziert werden.

COTANGENS

Mathematische Funktion, die meist nicht explizit im BASIC enthalten ist. Die Funktion COTANGENS gehört zu den trigonometrischen Funktionen, sie beschreibt das Verhältnis des *SINUS eines Winkels zum *COSINUS des gleichen Winkels. Allgemein ist der COTANGENS folgendermaßen definiert:

COTAN(X) = SIN(X)/COS(X) oder
COTAN(X) = 1/TAN(X)

Das BASIC des COMMODORE 64 enthält die COTANGENS Funktion nicht, sie kann aber leicht eingeführt werden und zwar mit Hilfe des *DEF FN-Statements.

Beispiel :

```
10 DEF FNCT(X) = 1/TAN(X)
20 INPUT "WINKEL";W
30 ERGEBNIS = FNCT(X*(ATN(1)/45))
```

```
40 PRINT "DER COTANGENS VON ";W;"IST : ";ERGEBNIS
50 GOTO 20
```

Es ist hierbei zu beachten, das die Winkel vom Computer im Bogenmaß dargestellt werden und nicht wie üblich im Gradmaß. Damit das Ergebnis sich auf Gradmaßwinkel bezieht, müssen die Winkel mit ATN(1)/45 multipliziert werden.

CP/M

Weitverbreitetes *Betriebssystem für *Mikrocomputer, das aus drei Komponenten besteht und zwar aus:

1. *CCP
2. *BIOS
3. *BDOS

CP/M ist heute fast zum Standard-Betriebssystem geworden, es zeichnet sich vor allem durch seine Zuverlässigkeit aus. Dieses Betriebssystem ist für den COMMODORE 64 durch zwei Erweiterungen auch nutzbar, es werden dafür die CP/M-Karte und die 80 Zeichen-Karte benötigt. Der Vorteil des CP/M Betriebssystems gegenüber dem DOS des COMMODORE 64 ist das große *Softwareangebot für CP/M. Für den Anwender des COMMODORE 64 ist dies in so weit ein Vorteil, weil er auf eine größere Auswahl von *Software zurückgreifen kann, für den Programmierer liegt der Vorteil darin, daß seine Programme einem viel größeren Anwenderkreis zugänglich sind.

CPU

Abkürzung für "CENTRAL PROCESSING UNIT", dies ist die Zentraleinheit eines *Computers, sie steuert die Peripheriegeräte, führt die Maschinenbefehle aus und

verarbeitet ,gemäß dem aktuellen Programm, die Daten. Die CPU ist quasi der Kern eines Computers.

Crash

Wörtlich übersetzt heißt dies "Zusammenbruch". Ist es nicht mehr möglich mit dem COMMODORE 64 zu arbeiten, weil er auf keinen Tastendruck reagiert und ändert sich dieser Zustand nicht, wenn er aus- und wieder eingeschaltet wurde, so kann die Ursache ein Crash sein.

CSNG

Befehl des MIKROSOFT-BASIC, der Variable beliebigen numerischen Typs in Variable einfacher Genauigkeit umwandelt.

SYNTAX :

CSNG (Variablenname)

Das BASIC des COMMODORE 64 enthält diesen Befehl nicht, aber es ist möglich ihn zu simulieren und zwar dadurch, daß statt einer Variablen zwei eingeführt werden. Der COMMODORE 64 kennt nur zwei Typen numerischer Variablen und zwar den *INTEGER-Typ und den Typ einfacher Genauigkeit. Es geht nun also darum eine Variable vom Typ *INTEGER in eine Variable einfacher Genauigkeit umzuwandeln. Diese Umwandlung soll an dem folgenden Beispiel gezeigt werden :

Beispiel :

```
10 INPUT A
20 INPUT B
30 X% = A/B
```

```
40 X = X%
50 X = X/A
60 ...
70 ...
```

In diesem Beispiel wurde die Umwandlung in Zeile 40 vorgenommen, so daß jetzt die Variable X die Stelle der Variablen X% eingenommen hat. Bei dem Befehl CSNG ist es nicht nötig die Kennung hinter dem Variablennamen zu ändern, da der COMMODORE 64 diesen Befehl nicht kennt, muß mit zwei Variablen unterschiedlicher Kennung gearbeitet werden.

CTRL

Teil der Tastatur des COMMODORE 64. Die Aufgabe dieser Taste ist es die Bildschirmausgabe zu verlangsamen. Wird zum Beispiel ein Programm durch das Statement *LIST ausgegeben und wird gleichzeitig die CTRL Taste betätigt, so läuft das Programmlisting erheblich langsamer auf dem Bildschirm ab. Diese Taste hat keinen *Tastaturcode und keinen *ASCII-CODE.

Cursor

Blinkendes Quadrat, das angibt an welcher Stelle des Bildschirms weitergeschrieben werden kann. Der Cursor kann mit den *Cursorsteuertasten frei über den Bildschirm bewegt werden, so daß jede Stelle erreicht werden kann und es möglich ist zu korrigieren einzufügen oder neu zu schreiben.

Cursorposition

Der COMMODORE 64 läßt es zu, die aktuelle Cursorposition immer genau zu bestimmen und den Cursor nach Wunsch zu plazieren. Dies ist nötig, um zum Beispiel komfortabele *INPUT-Routinen zu erstellen.
Die aktuelle Cursorposition läßt sich ohne weiteres aus zwei Speicherstellen der *Zeropage ablesen, in der einen Speicherstelle ist die aktuelle Zeile festgehalten, in der anderen die aktuelle Spalte. Durch die Abfrage dieser Werte mit *PEEK kann leicht die augenblickliche Position des Cursors ermittelt werden.

Beispiel :

```
PRINT "DER CURSOR STEHT IN ZEILE ";PEEK(214);"  SPALTE
";PEEK(211)
```

Die Speicherstellen, in denen die Zeile und die Spalte der Cursorposition festgehalten werden sind 214 für die Zeile und 211 für die Spalte.
Genauso leicht wie die Cursorposition ermittelt werden kann, kann der Cursor auch plaziert werden. Dafür müssen nur die Werte der Speicherstellen geändert werden und zwar durch *POKE, sowie einen *SYS-Befehl. Der SYS-Befehl ist nötig, da das *Betriebssystem noch die erforderlichen Zeiger für Bildschirm- und Farb-RAM berechnen muß, dafür steht aber im *Betriebssystem eine Routine zur Verfügung, die mit dem Befehl SYS 58640 angesprochen wird. Die benötigten POKE-Befehle haben folgende Struktur :

```
POKE 214,Zeile  bzw. POKE 211,Spalte
```

Cursorsteuerzeichen

Der *Cursor kann direkt aus dem Programm heraus geteuert werden und zwar innerhalb von *PRINT-Statements durch die

Cursorsteuerzeichen. Dabei ist zu beachten, das der Cursor nur in die Richtungen oben, unten, rechts und links gesteuert werden kann.

Steuerzeichen	Darstellung	Richtung
⇑		oben
⇓		unten
⇒		rechts
⇐		links

Es ist zulässig mehr als ein Cursorsteuerzeichen in ein *PRINT-Statement aufzunehmen. Die Taste mit den Cursorsteuerzeichen no nu hat den *Tastaturcode 7, die taste mit den Steuerzeichen nl , nr hat den *Tastaturcode 2.

CVD

Befehl des MIKROSOFT-BASIC, der die Aufgabe hat Zeichenkettenwerte, die nach einer *GET-Anweisung im Datenpuffer stehen, in mumerische Werte doppelter Genauigkeit umzuwandeln.

SYNTAX :

CVD (Variablenname)

Dieser Befehl ist im BASIC des COMMODORE 64 nicht enthalten und kann auch nicht simuliert werden, da der COMMODORE nicht mit numerischen Variablen doppelter Genauigkeit arbeitet.

CVI

Befehl des MIKROSOFT-BASIC, die Aufgabe dieses Befehls ist es, eine *Stringvariable, die nach einem *GET-Statement im

Datenpuffer steht, in eine numerische Variable vom Typ *INTEGER umzuwandeln.

SYNTAX :

CVI (Variablenname)

Dieser Befehl ist im Sprachumfang des COMMODORE 64-BASIC nicht enthalten, er kann aber simuliert werden.

Beispiel :

```
10 GET A$
20 IF ASC(A$)<48 OR ASC(A$)>57 THEN ...
30 A% = VAL(A$)
...
```

Durch diesen Programmteil werden Zeichen von der Tastatur eingelesen, falls eine Zahl eingegeben wird, wird sie auf eine *INTEGER-Variable zugewiesen.

CVS

Befehl des MIKROSOFT-BASIC, der Stringvariable, die nach einem *GET-Statement im Datenpuffer stehen, in numerische Variable einfacher Genauigkeit umwandelt.

SYNTAX :

CVS (Variablenname)

Das BASIC des COMMODORE 64 enthält diesen Befehl nicht, er kann aber simuliert werden.

Beispiel :

```
10 GET A$
20 IF ASC(A$)<48 OR ASC(A$)>57 THEN ...
30 A = VAL(A$)
...
```

Hier wird ein Zeichen von der Tastatur eingelesen, falls es eine Zahl ist, wird sie auf eine Variable einfacher Genauigkeit zugewiesen.

D

1. Ziffer des Hexadezimalsystems mit dem dezimalen Wert 13.

2.Taste der COMMODORE 64 Tastatur. Der *Tastaturcode dieser Taste ist 18. Außer dem Buchstaben "D" können noch der Buchstabe "d" im Groß-/Kleinschreibmodus, Das Graphikzeichen ◪ ,in Zusammenhang mit der Taste *SHIFT und das Graphikzeichen ⊟ in Zusammenhang mit der *COMMODORE Taste, erzeugt werden.
Diese Zeichen haben die folgenden *ASCII-CODES :

	Großschrift/Graphik	Groß-/Kleinschrift
D	68	196
d	-	68
◪	172	172
⊟	100	-

Diesen Zeichen werden die folgenden *Bildschirmcodes zugeordnet :

	Großschrift/Graphik	Groß-/Kleinschrift
D	4	68
d	-	4
◪	108	108
⊟	68	-

Die Umschaltung vom Großschrift/Grapik Modus auf den Groß-/Kleinschrift Modus kann durch die gleichzeitige Betätigung der Tasten *SHIFT und *COMMODORE vorgenommen werden.

Daisy chain

Verkettung von *Strings oder von *Dateien. Durch die Verkettung zweier Strings entsteht ein neuer String, während die beiden Ursprungstrings erhalten bleiben. Gleiches gilt auch für Dateien.

Daisy-wheel printer

*Typenraddrucker, als Typenraddrucker können auch bestimmte elektrische Schreibmaschinen verwendet werden.

DATA

Statement, das in allen BASIC-Dialekten vorhanden ist, es kennzeichnet *Daten die durch ein *READ-Statement gelesen werden sollen.

SYNTAX :

DATA Zahlen und/oder Strings

Es ist hierbei zu beachten, daß die einzelnen Angaben durch Komma voneinander getrennt werden und hinter der letzten Angabe kein Komma mehr steht.

Beispiel :

```
10 FOR I = 1 TO 5
20 READ A
30 PRINT A, SQR(A)
40 NEXT A
50 DATA 2,16,12,38,6
```

In diesem Beispiel werden die gelesene Zahl und ihre Quadratwurzel auf dem Bildschirm ausgegeben. Die Zahlen in der DATA-Zeile wurden zufällig gewählt, es können ebenso statt numerischer Daten auch alphanumerische Daten benutzt werden, es muß dann aber die Variable, in die die Daten eingelesen werden eine *Stringvariable sein.

Data Base

*Datenbank, eine Datenbank kann als Ansammlung von Dateien bezeichnet werde.

Data block

Datenblock, Anzahl von Datensätzen, die von einem *Magnetband eingelesen werden.

Data bus

*Datenbus, *Bus der der Datenübertragung dient.

Data communication

Datenübertragung zwischen zwei oder mehr Computern bzw. zwischen einem Computer und seiner *Peripherie.

Data compression

Datenverdichtung, darunter versteht man das *Packen von Daten.

Data format

Datenformat

Data input

Dateneingabe, Eingabe von Daten in einen Computer und zwar entweder über die Tastatu oder durch externe Massenspeicher.

Data input bus

Dateneingabebus, *Bus, der die Aufgabe hat die Dateneingabe von einem externen Massenspeicher zum Computer vorzunehmen.

Data medium

*Datenträger, dies könne zum Beispiel Hilfsmittel zur externen Speicherung von Daten sein.

Data movement

Datenverschiebung

Data output

Datenausgabe, darunter wird die Ausgabe von Daten auf eine Peripheriegerät verstanden, so ein Gerät kann ein *Massenspeicher, ein Drucker oder der Bildschirm sein.

Data processing

*Datenverarbeitung

Data rate

Datenrate, Maß der übertragenen Daten pro Zeiteinheit.

Data representation

Datendarstellung, Art und Weise, in der die Daten intern dargestellt werden, also nach welchem *Code sie verschlüsselt werden.

DATASETTE

Externer *Massenspeicher für den COMMODORE 64. Als Speichermmedium werden normale Casetten benötigt. Die DATASETTE ist der preiswerteste Massenspeicher für den COMMODORE 64. Bei der Auswahl der Cassetten sollte darauf geachtet werden, daß sie möglichst eine Spielzeit von weniger als 60 Minuten haben. Bei höheren Spielzeiten werden die Magnetbänder zu dünn, so daß es zu unerwünschten Effekten

kommen kann. Solche Effekte kommen dadurch zu Stande, daß die Bänder so stark magnetisiert werden,, daß auch die tiefer liegenden Teile magnetisiert werden, so daß es zu einem "Programmsalat" kommen kann. Außerdem sollten ruhig qualitativ hochwertige Bänder verwendet werden, was zu einer hohen Datensicherheit führt.

Data sink

Datenempfänger, dies kann jedes Gerät innerhalb eines EDV-Systems sein.

Data source

Datenquelle, Gerät von dem Daten gesendet werden.

Data word

Datenwort

Datei

Daten ,die sich gleichen oder das selbe Objekt beschreiben, werden zu einer Datei zusammengefaßt. Bei einer Kundendatei enthält ein Satz der Datei alle wichtigen Daten über einen Kunden, wie zum Beispiel Name, Wohnort, Telefonnummer, Umsatz, Datum des letzten Besuchs usw. Man kann eine Datei auch als das elektronische Gegenstück zur althergebrachten Kartei bezeichnen. Im Gegensatz zum *Array brauchen die Daten einer Datei nicht vom gleichen Typ zu sein, sie können sowohl numerisch, als auch alpanumerisch sein. Je nach der

Zugriffsart auf einzelne Datensätze unterscheidet man zwischen *sequentiellen und *relativen Dateien.

Datei, relative

Eine relative *Datei zeichnet sich dadurch aus, daß auf einen einzelnen Datensatz direkt zugegriffen werden kann, ohne daß vorher die ganze Datei eingelesen werden muß. Es ist also möglich, aus einer alphabetisch sortierten Telefondatei mit zum Beispiel 200 Sätzen den Datensatz von Herrn Müller herauszugreifen. Relativ organisierte Dateien lassen sich nur mit mindestens einem Diskettenlaufwerk realisieren.

Datei, sequentielle

Darunter versteht man eine *Datei, auf deren Sätze nicht direkt zugegriffen werden kann, sondern erst die ganze Datei gelesen werden muß. Alle Dateien, die sich auf einer Cassette befinden bzw. mit Hilfe der *Datasette abgespeichert wurden, sind sequentielle Dateien.

Dateischutz

Verfahren, um *Dateien vor unerlaubter Benutzung bzw. unbeabsichtigtem Löschen zu schützen. Ein solches Verfahren kann zum Beispiel durch ein *Password realisiert werden.

Dateiverarbeitung

Darunter versteht man die Benutzung und die Auswertung von

*Dateien. Eigene Programme zur Dateiverwaltung zu erstellen ist in der Regel sehr mühsam, da es sehr viele Nebenbedingungen zu beachten gilt, mittlerweile kann aber schon auf fertige *Software zurückgegriffen werden, wie zum Beispiel auf DATAMAT.

Daten

Dies sind Ziffern, Zahlen, Zeichen oder Worte, die mit Hilfe eines *Programms verarbeitet werden. Sie dienen der Darstellung von Informationen.

Daten, analoge

Ein klassisches Beispiel ist die Uhrzeit, wie sie zum Beispiel auf einer Kirchturmuhr angezeigt wird, oder die Geschwindigkeit, die von einem Tachometer angezeigt wird. Analoge Daten werden immer durch kontinuierliche Funktionen dargestellt. Die hervorstechendste Eigenschaft analoger Daten ist, daß sie unendlich fein abgestuft sind. Im Prinzip kann man sagen, daß die Uhrzeit von einer Analoguhr exakt dem Zeitablauf entsprechend angezeigt wird, während sie von einer Digitaluhr in Sekundensprüngen dargestellt wird. Analoge Daten bilden den Vorgang, den sie beschreiben, unendlich fein ab, während digitale Daten dies nur innerhalb bestimmter Intervalle tun.

Daten, digitale

Digitale Daten bestehen nur aus Zeichen. Die wichtigste Eigenschaft digitaler Daten ist, daß sie nur innerhalb vorgewählter Intervalle abgebildet werden. Die Uhrzeit wird

zum Beispiel von einer Digitaluhr in Intervallen von einer Sekunde angezeigt.

Daten, entpacken von

Umwandlung von *gepackten Daten in Daten normalen Formats. Diese Umwandlung kann mit dem COMMODORE 64 nicht vorgenommen werden, da der COMMODORE 64 das packen von Daten nicht vorsieht.

Daten, gepackte

Normalerweise werden Ziffern und Zeichen in einem *Byte verschlüsselt. Um die Ziffern 0-9 zu verschlüsseln werden nur 4 *Bits, also ein Halbbyte, benötigt. Die andere Hälfte des Bytes ist bei allen Ziffern identisch. Um *Speicherplatz zu sparen ist man dazu gekommen jeweils zwei Ziffern in einem Byte zu verschlüsseln. Die Umwandlung von der herkömmlichen Darstellung zu dieser neuen Darstellung nennt man Packen von Daten, das Ergebnis dieser Umwandlung sind dann gepackte Daten. Hierbei ist zu beachten, daß nur numerische Daten gepackt werden können, da zur Verschlüsselung von Buchstaben und Sonderzeichen ein ganzes Byte benötigt wird.

Datenbank

Sammlung von Dateien, geordnet nach Fachgebieten, mit der Möglichkeit, daß Benutzer bestimmte Daten nach gewissen Kriterien abfragen können. Ein Musterbeispiel einer Datenbank ist die Verkehrssünderkartei in Flensburg. Die Kommunikation zwischen Datenbank und Benutzer wird mit Hilfe bestimmter Datenbanksprachen, wie zum Beispiel d-BASE, abgewickelt.

Datenbus

Beim COMMODORE 64 ein Bündel von 8 Leitungen, das die Aufgabe hat die einzelnen Systemkomponenten (Prozessor, VIC-Chip, Sound-Chip, Periepheriebausteine usw.) miteinander zu verbinden und den Datentransfer zwischen diesen Komponenten vorzunehmen.

Datenerfassung

Unter Datenerfassung versteht man den Vorgang des Datensammelns, bevor die Daten in einem *Programm verarbeitet werden. Datenerfassung in einem Programm wird sehr häufig durch spezielle *Masken vorgenommen, die sogenannten Erfassungsmasken. Datenerfassung kann ohne weiteres auch durch einfach INPUT-Routinen vorgenommen werden. Einen Erfassungsmaske kann zum Beispiel folgendes Aussehen haben :

```
****************************************
*                                      *
* NAME        :                        *
*                                      *
* VORNAME     :                        *
*                                      *
* STR         :                        *
*                                      *
* PLZ         :                        *
*                                      *
* ORT         :                        *
*                                      *
* TELEFON     :                        *
*                                      *
* GEBURTSTAG  :                        *
*                                      *
****************************************
```

Diese Erfassungsmaske könnte zum Beispiel Bestandteil eines Programms zur Adressenverwaltung sein.
Eine einfach *INPUT-Routine könnte beispielsweise so aufgebaut sein:

```
10 PRINT CHR$(147)
20 INPUT "NAME        : ";N$
30 INPUT "VORNAME     : ";V$
40 INPUT "STR         : ";S$
50 INPUT "PLZ         : ";P%
60 INPUT "ORT         : ";O$
70 INPUT "TELEFON     : ";T$
80 INPUT "GEBURTSTAG  : ";G$
90 ...
100...
```

Diese INPUT-Routine dient ebenfalls der Erfassung der Daten, die für eine Adressdatei benötigt werden. Hierbei ist zu beachten, daß für die Erstellung einer Erfassungsmaske die

Mittel Feststellung und Festlegung der *Cursorposition benötigt werden.

Datenferübertragung

Hierbei werden Daten zum Beispiel von mobilen Erfassungsstationen erfasst und mittels eines *Akkustikkopplers an den Hauptrechner übertragen. Die Datenfernübertragung hat heute schon bei den Buchhandlungen eine große Verbreitung gefunden, sie sind mittels der Datenfernübertragung mit ihrem Großhändler verbunden und geben ihre Bestellungen direkt an den Computer des Großhändlers weiter.

Datenfluss

Darunter wird der Weg der Daten durch ein Programm verstanden. Dieser Weg wird an Hand eines *Datenflussplanes entworfen. Grundsätzlich besteht der Datenfluss aus drei Stationen und zwar 1. der Eingabe, 2. der Verarbeitung und 3. der Ausgabe.

Beispiel :

Es soll ein Programm zur Flächenberechnug erstellt werden. Der erste Schritt ist dann die *Datenerfassung, hier werden die benötigten Daten eingegeben. Der zweite Schritt ist die Verarbeitung der Daten, die Verarbeitung kann durch Rechnen mit den Daten, oder durch das Sortieren der Daten usw. geschehen. Der dritte Schritt ist schließlich die Ausgabe der Daten, das heißt es werden in diesem Beispiel die errechneten Flächen ausgegeben und zwar entweder über den Bildschirm, über einen Drucker oder über das Diskettenlaufwerk.
Bevor mit dem Programmieren begonnen wird, sollte der

Programmierer sich Gedanken über den Datenfluss innerhalb seines Programmes machen. Sinnvoll wäre es einen *Datenflussplan aufzustellen.

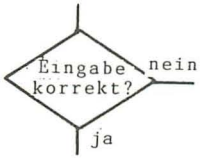
Datenflussplan

Hilfsmittel zur Erstellung von Programmen. Hier wird der Datenfluß des Programms weitgehend festgelegt. Diese Darstellungsart ist am weitesten verbreitet, da sie sehr leicht verständlich und übersichtlich ist, in den Fachzeitschriften (z.b. CHIP, MC usw.) finden sich regelmäßig solche Diagramme. Der Datenflussplan ist weitgehen identisch mit dem Programmablaufplan. Die Symbole dieser Pläne sind nach DIN 66001 genormt. Im Folgenden sollen einige dieser Symbole dargestellt werden.

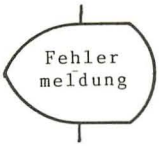
Programmstart bzw. Programmende

Anweisungen

Programmverzweigungen

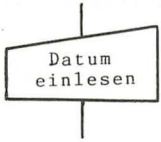


Bildschirmausgabe



Ein-/Ausgabe

Manuelle Eingabe



Listenausgabe

Ausgabe auf Diskette

Unterprogrammaufruf

Um den Datenfluß deutlich zu machen, ist es möglich die Daten innerhalb diese Diagramms mit einem anders farbigen Stift einzutragen.

Datenformat

Das Datenformat gibt an mit welcher Wortgröße die Daten in der CPU dargestellt und verarbeitet werden. Der COMMODORE 64 hat eine Wortgröße von 8 Bit, das heißt alle Befehle und Zeichen werden in einem *Byte codiert.

Datensatz

Eine *Datei besteht aus einer bestimmten Anzahl von Datensätzen. Eine Datei kann mit einer Kartei verglichen werden, die einzelnen Datensätze sind dann mit den einzelnen Karteikarten vergleichbar. Als Beispiel soll hier eine Adressdatei angeführt werden. Sie entspricht einer Adresskartei, nur mit dem Unterschied, daß sie sich nicht auf Papier befindet, sondern auf einer Diskette oder auf einer Cassette. Die einzelnen Datensätze dieser Datei könnten zum Beispiel aus folgenden Elementen bestehen :

1. Name
2. Voname
3. Straße
4. Posteitzahl
5. Ort
6. Telefonnummer

Ein Datensatz ist also die logische Zusamenfassung von Beriffen oder Daten, die zusammengehören, sowie Bestandteil einer *Datei.

Datenträger

Medium zur Speicherung und zum Transport von Daten. Die gebräuchlichsten Datenträger sind die Magnetplatte, das Magnetband und die Disketten. Weiter Datenträger, die früher häufiger verwendet wurden, sind die Lochkarte und der Lochstreifen.
Datenträger, die für den COMMODORE 64 nutzbar sind, sind die Diskette und das Magnetband in Form einer Audio-Cassette.

Datentypen, Umwandlung von

Der COMMODORE 64 kennt drei Datentypen und zwar die *Strings, die *Integer-Daten und die numerischen Daten einfacher Genauigkeit. Eine unzulässige Umwandlung ist die direkte Umwandlung von Strings in Integer-Daten oder in numerische Daten einfacher Genauigkeit. Diese Umwandlung kann nur über einen kleinen Umweg erreicht werden und zwar wie folgt :

A = VAL(A$) für die Umwandlung eines Strings in eine Variable einfacher Genauigkeit

und

A% = VAL(A$) für die Umwandlung eines Strings in eine Integer-Variable

Die Umwandlung zwischen den numerischen Datentypen läßt sich hingegen problemlos vornehmen und zwar :

A = A% für die Umwandlung einer Integer-Variablen in eine Variable einfacher Genauigkeit

und

A% = A für die Umwandlung einer Variablen einfacher Genauigkeit in eine Integer-Variable.

Datenübertragung

Darunter versteht man den Datenaustausch zwischen den Komponenten eines Systems, zwischen verschiedenen Systemen und zwischen *Prozessor und Peripherie. Ein Spezialfall der Datenübertragung ist die *Datenfernübertragung. Die Datenübertragung zwischen Prozessor und Peripherie kann auch als Datenein- und Datenausgabe bezeichnet werden.

Datenübertragung, parallele

Die Daten werden über 8 Datenleitungen vom Rechner zur Peripherie und umgekehrt übertragen. Hier werden 8 *Bits gleichzeitig übertragen.

Datenübertragung, seriell

In diesem Fall werden die Daten über eine Leitung bitweise übertragen. es wird jeweils nur ein *Bit übertragen und nicht mehrere Bits gleichzeitig. Das Gegenteil der seriellen Datenübertragung ist die *parallele Datenübertragung.

Datenverarbeitung

Hierunter werden alle Vorgänge, die sich mit der Sammlung und Auswertung von Daten befassen, vertstanden. Datenverarbeitung muß nicht unbedingt im Zusammenhang mit Computern geschehen, der Aufbau einer Kundenkartei mit normalen Karteikarten, ist ebenfalls eine Form der Datenverarbeitung. Wird Datenverarbeitung mit Computern betrieben, so spricht man von Elektronischer Datenverarbeitung kurz EDV.

Datenverarbeitung, kommerzielle

Dies ist eine spezielle Form der Datenverarbeitung und zwar wird dann von kommerzieller Datenverarbeitung gesprochen, wenn die Datenverarbeitung im Dienste der Wirtschaft steht. Beispiele für kommerzielle Datenverarbeitung sind :

1. EDV gestützte Finanzbuchhaltungssysteme

2. EDV gestützte Kostenrechnungssysteme
3. EDV gestützte Lohn- und Gehaltsabrechnung usw.

Dead time

Totzeit, das ist die Zeit, die nicht genutzt werden kann.

Debugger

Hilfsprogramm, das der Fehlerbeseitigung dient. Sehr häufig wird von diesem Programm der Wert der einzelnen Variablen ausgegeben und zwar immer dann, wenn das Programm auf die Variable zugreift. Der Programmierer kann den Datenfluss durch das Programm dann sehr gut verfolgen.

Debugging

Fehlerbeseitigung in einem Proramm, wörtlich übersetzt heißt debugging "Entkäferung"

Decimal to binary conversion

Dezimal-binär Umwandlung, darum handelt es sich immer dann, wenn eine Dezimalzahl in eine *Binärzahl umgewandelt wird.

Decision box

Entscheidungssymbol des Programmablaufplans, das eine *Verzweigung kennzeichnet

Decrement

Verminderung, ein *Zähler braucht nicht nur erhöht zu werden, es kann auch rückwärts gezählt, also vermindert werden.

Dedicated

Fest zugeordnet

DEF FN

Statement praktisch aller BASIC-Versionen, seine Zweck ist es dem Bennutzer die Möglichkeit zu geben, eigene Funktionen zu definieren, so daß lange Formeln bzw. lange Ausdrücke nur einmal im Programm ausführlich geschrieben werden müssen.

SYNTAX :

DEF FNname(Variable) = Formel bzw. Ausdruck

Hierbei ist zu beachten, daß der Name, der dem FN folgt, den Konventionen numerischer Variablennamen entsprechen muß. Weiterhin ist wichtig, daß die Definition eigener Funktionen nur für mathematische Funktionen zulässig ist.

Beispiel :

```
10 PRINT CHR$(147)
20 DEF FNFO(A) = SQR(A^2 + 5^2)
30 INPUT "SEITE A";A
40 B = FNFO(A)
50 PRINT "DIE LAENGE DER HYPOTHENUSE BETRAEGT ";B;"
EINHEITEN"
60 GOTO 30
```

Dieses Programm berechnet die Grundseite in einem rechtwinkeligen Dreieck, von dem bekannt ist, daß eine seiner Seiten 5 Einheiten lang ist. Die Länge der anderen Seite wird eigegeben und daraus anschließend die Länge der Hypothenuse mittels der selbst definierten Funktion berechnet.

DEF USR

Funktion des MIKROSOFT-BASIC, die es in dieser Form nicht im BASIC des COMMODORE 64 gibt. Die Funktion DEF USR läßt es zu, bis zu 10 Maschinenroutinen in einem Programm zu definieren.

SYNTAX :

DEF USR Nummer = INTEGER-Ausdruck

Das BASIC des COMMODORE 64 bietet nur die Möglichkeit eine Maschinenroutine von vornherein zu definieren bzw. aufzurufen.

DELETE

Befehl des MIKROSOFT-BASIC, der die Aufgabe hat einzelne Programmzeilen bzw. eine Gruppe von Programmzeilen zu

löschen.

SYNTAX :

DELETE Anfangswert - Endwert

Als Anfangs- und Endwert werden hier die Zeilennummern angegeben, die gelöscht werden sollen. Dieser Befehl existiert im BASIC des COMMODORE 64 nicht. Hier müssen die Zeilen einzeln von "Hand" gelöscht bzw. überschrieben werden.

Development system

Entwicklungssystem, das zur Entwicklung von *Software dient.

Development time

Entwicklungszeit, das ist die Zeit, die benötigt wird um ein Programm zu entwickeln.

Dezimalsystem

*Zahlensystem, dessen Basis die Zahl 10 ist. Dieses Zahlensystem ist das gebräuchlichste im Alltagsleben. *Computer rechnen gewöhnlich im *Binären System, dessen Basis die Zahl 2 ist. Ein weiteres gebräuchliches System ist das mit der Basiszahl 16, das *Hexadezimalsystem.

Dialog

Darunter wird die Kommunikation zwischen *Benutzer und *Computer verstanden. Der Dialog besteht darin, daß der Computer auf die Eingaben des Benutzers mit Meldungen oder Kommentaren reagiert. Eine solche Reaktion kann zum Beispiel eine Fehlermeldung sein, falls *Daten eingegeben wurden, die unzulässig sind.

Dialogbetrieb

Es gibt mehrere Möglichkeiten mit einem *Computer zu arbeiten und zwar den *Stapelbetrieb und den Dialogbetrieb. Im Dialogbetrieb ist es üblich an einem Bildschirm zu arbeiten, auf dem die Nachrichten des Computers ausgegeben werden. Der Dialogbetrieb wird vor allen Dingen beim Ablauf anwenderorientierter Programme benutzt. Die Eingaben des *Benutzers werden durch *Masken,die sehr genaue Anweisungen enthalten können, unterstützt. Der COMMODORE 64 bietet sich zum Dialogbetrieb geradezu an, durch seine hohe Verarbeitungsgeschwindigkeit und die Möglichkeit der *Cursorpositionierung gibt er die Hilfsmittel zur Maskenerstellung dem Programmierer an die Hand. Das augenfälligste Kennzeichen des Dialogbetriebs ist, daß der Benutzer die Möglichkeit hat direkt in den Programmablauf einzugreifen bzw. den Programmablauf zu unterbrechen.

Dialogsteuerung

Der *Dialogbetrieb von *Anwendungssoftware muß gesteuert werden, damit der *Benutzer nicht willkürlich durch das *Programm springen kann. Diese Steuerung ist am Besten durch die *Menuetechnik zu realisieren. Ein Beispiel für menuegesteuerte Anwendungssofware ist DATAMAT.

Dienstprogramm

Ein Dienstprogramm ist ein *Programm, das den Umgang mit dem *Computer leichter macht bzw. dem Programmierer Arbeit abnimmt. Ein solches Dienstprogramm ist zum Beispiel *PERFORMANCE TEST der Demo/Test-Diskette der COMMODORE Diskettenlaufwerke. Dieses Programm hat die Aufgabe alle Funktionen des Diskettenlaufwerks zu testen und eine Diskette zu *initialisieren und zu *formatieren. Dienstprogramme können vom Programmierer ohne weiteres selbst erstellt werden, zum Beispiel ein Programm mit dem Drucker an den *Computer angepaßt werden können. Unter den Begriff Dienstprogramm fallen auch solche Programme, die einmal geschrieben wurden und immer wieder benutzt werden, zum Beispiel *Sortierroutinen.

Digit

Ziffer, Stelle

Digital-Analog-Wandler

Gegenteil des *A-D-Wandlers. Der Digital-Analog-Wandler wandelt das Ergebnis einer digitalen Rechnung in ein analoges Ausgangssignal um.

Digitalrechner

Dies sind Rechner, die mit Zahlenwerten, das heißt mit

Ziffern, arbeiten. Diese Zahlenwerte werden mit Hilfe der vier Grundrechenarten verarbeitet, sie dürfen stets nur genau festgelegte Zustände annehmen. Jeder *Computer ist ein Digitalrechner, der einfachste Digitalrechner ist der Abakus. Am Beispiel des Abakus kann gezeigt werden, was unter genau festgelegten Zuständen zu verstehen ist. Die Kugeln des Abakus können nur zwei Zustände annehmen, entweder sie bleiben an ihrem Platz, oder sie werden verschoben, Zwischenzustände gibt es nicht. Ein Computer arbeitet praktisch nach dem gleichen Prinzip, entweder es fließt Strom, oder es fließt kein Strom, es existieren nur die Zustände 0 und 1.
Das Gegenteil eines Digitalrechners ist ein Analogrechner, dieser arbeitet nur mit stetigen physikalischen Größen.

Digitization

Digitalisierung, Umsetzung von *analogen Daten in *digitalen Daten.

DIM

Statement, das es in allen BASIC-Dialekten gibt. Es hat die Aufgabe *Speicherplatz für einen *Array zu reservieren. Arrays, die weniger als 11 Elemente enthalten brauchen nicht dimensioniert zu werden.

SYNTAX :

DIM Variablenname(Feldzahl 1, Feldzahl 2, ...)

Der COMMODORE 64 läßt es zu einen Array bis zu 255 Dimensionen anzulegen. Sinnvoll ist dies allerdings nicht, da solche Arrays absolut unübersichtlich sind, Dimensionierungen

bis zu drei oder vier Dimensionen sind noch gut durchzuführen und auch noch übersichtlich. Die DIM-Anweisung für einen Array, der aus 10 Feldern in drei Dimensionen bestehen soll, hat folgendes Aussehen:

```
DIM A(10,10,10)
```

Das BASIC des Commodore 64 bietet die Möglichkeit mehrere Arrays in einer Programmzeile zu dimensionieren und zwar wie folgt :

```
10 DIM B(5),C$(6,4),A(8,9,2)
```

Durch dieses Statement wurden drei Arrays dimensioniert und zwar 1.der Array B, der nur eine Dimension hat, die Länge, und fünf Felder enthält.

2. Der Array C$, der zwei Dimensionen hat, die Länge und die Breite, in der Länge enthält er 6 Felder und in der Breite vier Felder. Dieser Array hat folgende Gestalt:

```
Länge
        I 1 I 2 I 3 I 4 I Breite

  1     I   I   I   I   I
        -----------------
  2     I   I   I   I   I
        -----------------
  3     I   I   I   I   I
        -----------------
  4     I   I   I P I   I
        -----------------
  5     I   I   I   I   I
        -----------------
  6     I   I   I   I   I
        -----------------
```

Dieser Array ist als alphanumerische Variable definiert, er kann Buchstaben, Zahlen und Sonderzeichen enthalten. Es ist möglich direkt auf jedes einzelne Feld des Arrays zuzugreifen, das P im oberen Bild ist im Feld 4,3 des Arrays C$ hinterlegt. Für die Felder eines String-Arrays, wie C$, gelten die gleichen Regeln, wie für normale *String-Variablen. String-Arrays können zum Beispiel benutzt werden, wenn Adressdateien, Schallplattenregister, Bücherregister usw. erstellt werden sollen.

3. wird durch das Statement noch der Array A dimensioniert, der drei Dimensionen hat, Länge, Breite und Tiefe. In der Länge enthält dieser Array 8 Felder, in der Breite 9 Felder und in der Tiefe 2 Felder. Der Aufbau ist analog zu dem Aufbau des Arrays C$.
Für die Dimensionierung von Arrays ist es wichtig, daß die Größe der Dimensionen auch durch Variable festgelegt werden kann.

Beispiel :

```
10 INPUT X
20 DIM A(X)
30 ...
40 ...
```

Weiterhin muß beachtet werden, daß ein Array nur einmal in einem Programm dimensioniert werden darf, üblicherweise geschieht dies am Anfang des Programms. Die Dimensionierung muß vor dem ersten Zugriff auf den Array geschehen.

DIR

Hilfsprogramm der COMMODORE Test-/Demodiskette. Es bietet folgende Möglichkeiten :

d - Zeigt das Directory der eingelegten Diskette an

> - In Zusammenhang mit diesem Zeichen können die Diskettenbefehle in verkürzter Form eingegeben werden. Zum Beispiel >N:TEST,KN zum *Formatieren einer Diskette.

q - Dient zum Beenden des proggramms DIR

s - Zeigt den Fehlerkanal an.

Direct access

Direktzugriff, wird zum Beispiel auf einen Datensatz direkt zugegriffen, ohne daß vorher die ganze Datei gelesen werden muß, so spricht man von Direktzugriff.

Direct addressing

*Direkte Adressierung

Direct memory access

Direkter Speicherzugriff

Directory

Dateiverzeichnis, wird automatisch von jeder Diskette, die initialisiert und formatiert ist, angelegt. Das Directory kann durch folgenden Befehl angezeigt werden :

```
LOAD "$",8
LIST
```

Wurde das Directory auf diese Art und Weise geladen, so wird das im Hauptspeicher befindliche Programm gelöscht.

Direktzugriff

Das *DOS der COMMODORE Diskettenlaufwerke bietet die Möglichkeit auf jeden Block der Diskette direkt zuzugreifen. Jeder Block der Diskette ist durch die Nummer des Tracks und des Sektors genau bestimmbar. Der direkte Zugriff auf einzelne Blöcke bietet nun die Möglichkeit der Manipulation einzelner Files bis hin zum Aufbau einer eigenen Dateistruktur.
Dem *DOS muß mitgeteilt werden, daß im Direktzugriff

gearbeitet werden soll, dies geschieht durch folgenden Befehl:

OPEN 1,8,2 "#"

Mit diesem Befehl wird der logischen Filenummer 1 auf dem Gerät 8, dem Diskettenlaufwerk, eine Direktzugriffsdatei zugeordnet. Zur Übermittlung der Daten wird der Kanal 2 des Diskettenlaufwerks benutzt. Diese Angaben werden vom *DOS benötigt, da für den Direktzugriff ein Datenkanal und ein Datenpuffer vom *DOS zugeordnet werden müssen.
Als Kanalnummern sind die Zahlen 2 bis 14 zulässig, die Kanäle 0 und 1 sind für LOAD und SAVE reserviert, der Kanal 15 ist der Kommadokanal.
Für den Direktzugriff auf die Diskette stehen folgende Befehle zur Verfügung :

1. BLOCK-READ
2. BLOCK-WRITE
3. BLOCK-ALLOCATE
4. BLOCK-FREE
5. BLOCK-EXECUTE

Der Direktzugriff auf die Diskette bietet eine Vielzahl von Möglichkeiten :

1. Manipulation einzelner Sektoren
   Hiermit kann zum Beispiel der Diskettenname bzw. die ID geändert werden, oder einzelnen Dateien können neue Name gegeben werden.
2. Versehentlich gelöschte Programme können gerettet werden.
3. Aufbau einer eigenen Dateistruktur, die das DOS nicht kennt

Disabled

Gesperrt, es ist möglich Geräte und/oder Programme für unbefugte Benutzer zu sperren und zwar durch ein *Password.

Disjunktion

Operation der *Boolschen Algebra, sie verknüpft zwei Operanden durch OR. Der zu prüfende Ausdruck ist dann wahr, wenn einer der beiden Operatoren wahr ist.

DISK ADDR CHANGE

Hilfsprogramm der Test-/Demodiskette der COMMODORE Diskettenlaufwerke. Durch diese Programm können die Gerätenummern der angeschlossenen Diskettenlaufwerke geändert werden. Die zulässigen Gerätenummern liegen zwischen 4 und 15 nach dem Programmstart müssen die alte Gerätenummer und die neue Gerätenummer eingegeben werden, das Programm stellt die Gerätenummer dann auf den neuen Wert um. Sinnvoll anzuwenden ist dies Programm, wenn mehr als ein Diskettenlaufwerk betrieben werden soll.

Disk drive

Festplattenlaufwerk bw. Diskettenlaufwerk

Diskette

*Datenträger, der aus einer schwarzen Kunststoffscheibe

besteht, deren Oberfläche magnetisierbar ist. Die Kunststoffscheibe ist in einer Kartonhülle, die die Diskette vor mechanischen Einwirkungen schützen soll. Für die Arbeit mit dem COMMODORE 64 und einem COMMODORE Diskettenlaufwerk können nur 5.25 Zoll Disketten verwendet werden. Eine Diskette kann verschiedene Eigenschaften haben, *single sided, *double sided, *single density und *double density. Für die Arbeit mit dem Commodore 64 sind Disketten, die die Eigenschaften single sided, single density haben vollkommen ausreichend.

Diskette, Formatieren einer

Jede neue Diskette muß vor dem ersten Gebrauch formatiert werden, das heißt sie muß vorbereitet werden. Das Formatieren geschieht durch folgenden Befehl :

OPEN 1,8,15,"NEW:Diskettenname,Kennung"

Der Vorgang des Formmatierens dauert etwa 80 Sekunden. Dieser Vorgang wird von dem *Prozessor des Diskettenlaufwerks übernommen, so daß während des Formatierens mit dem COMMODORE 64 weitergearbeitet werden kann. Der Befehl NEW kann durch N abgekürtzt werden.

Diskette, Initialisieren einer

Eine Diskette initialisieren heißt, daß die Disketten *BAM in den Speicher des Diskettenlaufwerks geschrieben wird. Dieser Vorgang ist wichtig, wenn mit *Direktzugriffsbefehlen gearbeitet wird, da diese Befehle die im Diskettenspeicher liegende BAM lesen, um festzustellen welche Blöcke frei oder

belegt sind. Haben alle Disketten eine unterschiedliche Kennung, so erübrigt sich das Initialisieren, da das Diskettenlaufwerk durch die unterschiedliche Kennung feststellen kann, daß eine neue Diskette eingelegt wurde.

Der INITIALIZE-Befehl hat folgende Syntax :

```
OPEN 1,8,15,"I"
CLOSE 1
```

Diskettenlaufwerk

Gerät, das Daten und Programme von Disketten liest bzw. auf Disketten schreibt. Ein anderer Name für Diskettenlaufwerk ist auch Floppy-Disk. Im Zusammenhang mit dem COMMODORE 64 ist das gebräuchlichste Diskettenlaufwerk SINGLE FLOPPY DISK 1541. Diese Laufwerk bereitet die *Disketten so vor, daß sie die folgenden technischen Daten besitzen :

Anzahl Spuren	:	35
Anzahl der Sektoren pro Spur	:	17 bis 21
Bytes je Block	:	256
Gesamtzahl der Blöcke	:	683
Zahl der freien Blöcke	:	644
Einträge in das Directory	:	144

Die Differenz zwischen der Gesamtzahl der Blöcke und der Anzahl der freien Blöcke kommt dadurch zustande, daß das Directory 39 Blöcke belegt.
Die Gerätenummer des Diskettenlaufwerks ist standardmäßig 8, kannn aber auf Werte zwischen 4 und 15 geändert werden und zwar mit dem Hilfsprogramm *DISK ADDR CHANGE.
Es ist mit dem IEC-BUS möglich alle anderen COMMODORE Diskettenlaufwerke an den COMMODORE 64 anzuschließen.

Disk storage

Englisches Wort für Plattenspeicher, darunter wird die Speicherung auf eine *Festplatte oder eine *Diskette verstanden.

Displacement

*Distanzadresse

Display

Synonymer Ausdruck für Anzeige bzw. Anzeigegerät. Die Anzeige eines Taschenrechners ist beispielsweise ein Display. Ein *Monitor oder ein Bildschirm können ebenfalls als Display bezeichnet werden.

Display console

Bildschirmkonsole, darunter wird ein Bildschirmarbeitsplatz verstanden, der aus einem Bildschirm und einer Tastatur besteht.

Display register

Anzeigeregister

DISPLAY T&S

Dienstprogramm der COMMODORE Test / Demodiskette. Seine Aufgabe ist es den Aufbau der einzelnen Blocks einer Diskette auf dem Bildschirm oder dem Drucker auszugeben. Nachdem das Programm mit *RUN gestartet worden ist, werden die gewünschte Spur (TRACK) und der gewünschte Block (SECTOR) eingegeben, der Inhalt dieses Blocks wird dann entweder über den Drucker oder über den Bildschirm wieder ausgegeben.

Double Density

Diese Eigeschaft erlaubt es prinzipiell auf einer *Diskette die doppelte Datenmenge unterzubringen. Dies ist im Zusammenhang mit dem COMMODORE 64 aber uninteressant, da es für ihn zur Zeit nicht die erforderlichen Hardwareerweiterungen gibt.

Double sided

Disketten mit dieser Eigenschaft können auf beiden Seiten beschrieben werden. Theoretisch ist es möglich auch *single sided Disketten auf beiden Seiten beschrieben werden, nur ist in diesem Fall die Datensicherheit nicht auf beiden Seiten gewährleistet, da nicht beide Diskettenseiten getestet worden sind.

DOS

Betriebssystem des Diskettenlaufwerks. DOS ist die Abkürzung für "Disk operating system". Das Betriebssystem des Diskettenlaufwerks übernimmt die Steuerung der internen

Vorgänge des Diskettenlaufwerks und die Ausführung der Befehle, die vom Computer übermittelt werden.

Down time

Ausfallzeit

Drive

Laufwerk. Damit kann entweder ein *Diskettenlaufwerk der ein *Festplattenlaufwerk gemeint sein.

Drucker

Ein Drucker kann nur als *Ausgabegerät benutzt werden. Seine Aufgabe ist es Programmlistings auszugeben, Kopien des Bildschirminhaltes herzustellen oder auch Graphiken oder Listen zu erstellen. Hierbei ist zu beachten, daß nicht mit jedem Ducker Graphik erzeugt werden kann, er muß diese Fähigkeit ausdrücklich besitzen.

Dualsystem

Zahlensystem, dessen Basis die Zahl 2 ist, es kennt nur die Ziffern 0 und 1. Dieses Zahlensystem findet haupsächlich in der elektronischen Datenverarbeitung Anwendung. Ein Computer verschlüsselt alle Daten binär, das heißt alle Zahlen werden in das Dualsystem umgesetzt.
Die dezimalen Zahlen 1 - 10 werden im Dualsystem folgendermaßen dargestellt :

1	1	1*2^0
2	10	1*2^1+0*2^0
3	11	1*2^1+1*2^0
4	100	1*2^2+0*2^1+0*2^0
5	101	1*2^2+0*2^1+1*2^0
6	110	1*2^2+1*2^1+0*2^0
7	111	1*2^2+1*2^1+1*2^0
8	1000	1*2^3+0*2^2+0*2^1+0*2^0
9	1001	1*2^3+0*2^2+0*2^1+1*2^0
10	1010	1*2^3+0*2^2+1*2^1+0*2^0

Dynamic memory

*Dynamischer Speicher

Dynamic RAM

*Dynamisches RAM

E

1. Ziffer des *Hexadezimalsystems mit dem dezimalen Wert 14.

2. Taste des COMMODORE 64 mit dem Tastaturcode 14. Durch diese Taste können die Zeichen E, e, ⊞ , ⊟ dargestellt werden. Diese Zeichen haben die folgenden *ASCII-Codes :

	Großschrift/Grapik	Groß-/Kleinschrift
E	69	197
e	-	69
⊞	177	177
⊟	183	-

Neben dem ASCII-Code werden den Zeichen auch noch *Bildschirmcodes zugeordnet. Hierbei ist zu beachten, daß der COMMODORE 64 zwei Zeichensätze hat und zwar einmal den Großschrift-/Grapikzeichensatz und zum anderen den Groß-/Kleinschriftzeichensatz. Die Umschaltung von einem Zeichensatz auf den anderen wird durch gleichzeitige Betätigung der Tasten *SHIFT und *COMMDORE vorgenommen. Satz 1 ist der Großschrift-/Grapikzeichensatz, Satz 2 der Groß-/Kleinschriftzeichensatz.

Bildschirmcodes :

Zeichen	Satz 1	Satz 2
E	5	69
e	-	5
⊞	113	113
⊟	120	-

EBCDIC-Code

Dieser Code veschlüsselt alle Zeichen in einem *Byte. EBCDIC ist die Abkürzung für "Extended binary coded decimal interchange code". Dieser Code wird haupsächlich von der IBM und Siemens benutzt.

Echtzeitbetrieb

Art einen Computer zu betreiben. In diesem Fall werden die Daten sofort, wenn sie entstanden sind, verarbeitet. *Homecomputer und *Mikrocomputer arbeiten praktisch nur im Echtzeitbetrieb. Das Gegenteil des Echtzeitbetriebes ist die *Stapelverarbeitung.

Editieren

Darunter versteht man die Veränderung eines Programmtextes. Ist beim Programmablauf ein Fehler aufgetreten und soll er behoben werden, so muß das Programm editiert werden.

Editor

Teil des Betriebssystems, der das Editieren von Programmen unterstützt.
Der COMMODORE 64 hat einen sehr komfortabelen Editor. Es ist möglich den fehlerbehafteten Programmteil durch das Statement *LIST auszugeben, dann mit dem *Cursor in die fehlerhafte Zeile zu fahren, die Änderung durchzuführen, sie durch Betätigung der *RETURN Taste zu bestätigen. Auf diese Art und Weise können mehrere Änderungen durchgeführt werden. Der Vorzug des COMMODORE 64 Editors ist, daß bis zu 24 Zeilen auf

dem Bildschirm in einem Vorgang editiert werden können.

EDV

Abkürzung für elektronische Datenverarbeitung. Darunter versteht man die Verarbeitung von Daten unter zu Hilfe nahme von Computern und/oder anderen elektronischen Hilfsmitteln (Taschenrechner). Die EDV kann immer dann sinnvoll eingesetzt werden, wenn große Datenmengen analysiert, sortiert oder sonst wie verarbeitet werden müssen.

Ein-/Ausgabebus

Parallele Leitungen, die den Datentransfer von den Ein-/Ausgabeschnittstellen zum Hauptbus regeln.

Ein-/Ausgabegerät

Peripheriegeräte, die sowohl zur Dateneingabe, als auch zur Datenausgabe dienen. Solche Geräte sind die *Datasette und das *Diskettenlaufwerk. Ein reines Ausgabegerät ist der Drucker, reine Eingabegeräte sind der Lochkartenleser und der Lochstreifenleser. Ein weiters Ein-/Ausgabegerät ist das *Festplattenlaufwerk.

Ein-/Ausgabekanal

Bestimmte Zahl von Leitungen, die zum Datenverkehr zwischen dem Computer und den Ein-/Ausgabegeräten benutzt werden. In diesen Kanal kann die Steuerung der Datenübertragung schon

integriert sein.

Ein-/Ausgabeport

Schittstelle eines Computers, der ihn mit der Außenwelt verbindet, über diese Schnittstelle werden Daten sowohl ein- als auch ausgegeben.

Eingabemaske

Programmiertechnisches Hilfsmittel zur *Datenerfassung im *Dialogbetrieb. Eingabemasken können sowohl *Auswahlmasken als auch Datenerfassungsmasken sein. Ihre Aufgabe ist es dem Benutzer deutlich zu machen welche Daten von ihm erwartet werden.

Einlesen

Werden Daten oder Programme von einem Periepheriegerät in einen Computer transferiert, so werden sie eingelesen. Sehr deutlich wird dies am Beispiel eines Lochkartenlesers, der erst den gesamten Inhalt einer Lochkarte abtastet und ihn anschließend an den Computer weitergibt. Dieses Vorgehen findet sich auch in ähnlicher Weise bei der *Datasette und dem *Diskettenlaufwerk, diese Geräte lesen Zeichen für Zeichen die Information von den *Datenträgern ein und geben sie an denCOMMODORE 64 weiter.

Einplatzsystem

Darunter versteht man eine EDV-Anlage, also einen Computer mit Peripheriegeräten, an der nur eine Person arbeiten kann und nicht mehrere Personen gleichzeitig.

Electronic data processing

Elektronische Datenverarbeitung (EDV).

END

Statement, das in allen *BASIC-Dialekten vorkommt. Seine Aufgabe ist es das Programm zu beenden, alle Dateien zu schließen und in die *Commandoebene zurückzukehren.

SYNTAX :

END

Dieses Statement kann an jeder Stelle im Programm stehen, wird es im Programmablauf erreicht, so wird das Programm beendet. Hat ein Programm keine Unterprogramme, so kann auf das END-Statement verzichtet werden, das Programm wird dann beendet, wenn das letzte Statement abgearbeitet worden ist. Enthält ein Programm aber Unterprogramme, so muß das letzte Statement das Hauptprogramms das END sein, da sonst der Programmablauf mit dem ersten Unterprogramm fortgesetzt wird und es zu der *Fehlermeldung "RETURN WITHOUT GOSUB" kommt.
Ist ein Programm mit END abgeschlossen worden, so erscheint auf dem Bildschirm die Meldung "READY", das Programm kann dann durch *RUN oder durch *CONT neu gestartet werden.

Endlosschleife

Eine *Schleife, die von einem Programm unendlich oft durchlaufen wird, so daß das Programm niemals beendet wird.

Beispiel :

```
10 PRINT "COMMODORE 64"
20 GOTO 10
30 END
```

Dieses kleine Programm läuft praktisch unendlich lange, da es niemals zur Zeile 30 kommt und somit beendet wird. Es ist ein typisches Beispiel einer Endlosschleife.

EOF

Abkürzung für "End of File". Meist ist dies das letzte Zeichen eines Datenblocks, so daß vom Programm aus das Ende einer Datei erkannt werden kann. Der COMMODORE 64 bietet diese Möglichkeit nicht, sie muß durch einen kleinen Umweg erzeugt werden. Die einfachste Möglichkeit das Dateiende zu kennzeichnen ist, in die Datei als letztes Element einen Wert zu schreiben, der eigentlich in dieser Datei gar nicht vorkommen kann. Wird diese Datei später wieder eingelesen, so muß auf diesen unlogischen Wert hin abgefragt werden, ist der Wert gelesen worden, so ist das Dateiende erreicht.

EPROM

Dies ist ein Festwertspeicher, der programmiert und gelöscht werden kann. Programme ,die in einem EPROM enthalten sind werden nicht gelöscht, wenn der Strom abgeschaltet wird.

EPROMS können durch Bestrahlung mit ultraviolettem Licht gelöscht werden.

ERASE

1. Englischer Ausdruck für Löschen

2. Anweisung des MIKROSOFT-BASIC. Diese Anweisung hat die Aufgabe den Speicherplatz, der für *ARRAYS reserviert worden ist, wieder freizugeben.

SYNTAX :

ERASE (Variablenname)

Dieser Befehl existiert im BASIC des COMMODORE 64 nicht, er läßt sich auch nicht ohne großen Aufwand simulieren.

Erasable storage

Speicher, der löschbar ist.

Error detection

Fehlersuche, sowohl in einem Programm, als auch in einem Gerät.

Error message

*Fehlermeldung, diese werden immer dann abgesetzt, wenn ein

Fehler gemacht wurde, entweder im Programm, oder wenn zum Beispiel das Diskettenlaufwerk angesprochen wurde, obwohl es nicht eingeschaltet war.

Execution time

*Ausführungszeit, dies ist die Zeit, die ein Programm vom Start bis zum Ende braucht.

EXP

Schlüsselwort des COMMODORE-BASIC. Diese Funktion dient der Ermittlung von Potenzen der Eulerschen Zahl e.

SYNTAX :

EXP (Zahl oder Variablenname)

Wird zum Beispiel in diesen Ausdruck als Zahl 10 eingetragen, so ist das Ergebnis dieses Ausdrucks der Wert von e^10.

Beispiel :

```
10 INPUT "POTENZ ";X
20 Y = EXP (X)
30 PRINT "E^";X;" BETRAEGT ";Y
40 GOTO 10
```

Durch dieses Beispielprogramm können beliebige Potenzen der Eulerschen Zahl berechnet werden. Es ist hierbei zu beachten, daß die Potenz, zu der e genommen werden soll, nicht größer als 88 sein darf, da diese Funktion dann ein Ergebnis hat, das von dem COMMODORE 64 nicht mehr verarbeitet werden kann.

Expansion Port

Der Expansion Port ist die vielseitigste *Schnittstelle des COMMODORE 64. Grundsätzlich ist dieser Port durch ein BASIC Programm nicht zu bearbeiten, hier muß auf die Maschinensprache zurückgegriffen werden. Auf dem Expansion Port ist der gesamte *Systembus herausgeführt, eine unsachgemäße Behandlung des Expansion Ports kann hohe Reperaturkosten zur Folge haben. Der Vorteil des Expansion Ports liegt darin, daß hier praktisch jede Erweiterung angeschlossen werden kann, sei es nun eine BASIC-Erweiterung, eine andere *Programmiersprache, ein *Interface oder ein Steckmodul.

Exponentialschreibweise

Synonymer Ausdruck für wissenschaftliche Notation. Wenn sehr große oder sehr kleine Zahlen verarbeitet werden, ist es oft sehr mühsam die Größenordnung der einzelnen Zahlen fetszustellen, falls diese Zahlen in der üblichen Schreibweise dargestellt worden sind. Die Exponentialschreibweise stellt die Zahlen als Produkt einer Zahl, die zwischen 1 und 10 liegt, und einer Zehnerpotenz dar. Die Zahl 1984 zum Beispiel wird in der Exponentialschreibweise so dargestellt 1.984 * 10^3.
Dieses Vorgehen mag bei kleineren Zahlen unsinnig erscheinen, überschreiten die Zahlen aber die Millionengrenze, so wird das ganze schon sehr sinnvoll.

Beispiel :

123400000 = 1.234 * 10^8

Die Exponentialschreibweise bietet sich immer dann an, wenn Zahlen von der Größenordnung her miteinander verglichen werden sollen.

External storage

Externer Speicher, dies sind Speicher, die sich nicht innerhalb des Computers befinden, sondern auf *Peripheriegeräten, wie zum Beipiel einem *Diskettenlaufwerk oder der *Datasette.

F

1. Ziffer des *Hexadezimalsystems mit dem dezimalen Wert 15.

2. Taste der COMMODORE 64 Tastatur, mit der die Zeichen F, f, ◪ , ▭ dargestellt werden können. Diese Taste hat den *Tastaturcode 21. Die Zeichen, die durch diese Taste dargestellt werden können, haben die folgenden *ASCII-Codes :

	Großschrift/Graphik	Groß-/Kleinschrift
F	70	198
f	-	70
◪	187	187
▭	102	-

Diese Zeichen haben die folgenden *Bildschirmcodes :

	Großschrift/Graphik	Groß-/Kleinschrift
F	6	70
f	-	6
◪	123	123
▭	70	-

Die Umschaltung zwischen den beiden Modi kann durch gleichzeitige Betätigung der Tasten *SHIFT und *COMMODORE herbeigeführt werden.

F1 - F8

Funktionstasten der Tastatur des COMMODORE 64. Diese Tasten haben die * ASCII-CODES von 133 - 140. Die Tastaturcodes lauten wie folgt :

F1/F2	4
F3/F4	5
F5/F6	6
F7/F8	3

Der COMMODORE 64 biete die Möglichkeit diese Tasten mit selbstdefinierten Funktionen zu belegen.

Beispiel :

```
10 GET A
20 IF A = 133 Then  GOSUB 1000
...
...
1000 PRINT "COMMODORE 64"
```

In diesem Beispiel wird das Unterprogramm aufgerufen, wenn die Funktionstaste F1 betätigt wird. Eine sinnvolle Einsatzmöglichkeit der Funktionstasten ist zum Beispiel, wenn ein komplexes Programm vorliegt bei Betätigung einer bestimmten Funktionstaste Erläuterungen auf den Bildschirm zu senden.

Fakultät

Mathematische Funktion, die eine bestimmte Zahl von Faktoren miteinander multipliziert. Diese Funktion wird sehr häufig in der Statistik benötigt. Allgemein wird Fakultät so dargestellt :

n!

Für n kann eine beliebige Zahl eingesetzt werden. Diese Funktion berechnet dann den Wert der Multiplikation aller Zahlen von 1 bis n.

Beispiel :

5! = 1 * 2 * 3 * 4 * 5 = 120

Die Fakultät ist im BASIC-Sprachumfang nicht enthalten, sie kann durch ein kleines Programm aber leicht erzeugt werden.

Beispiel :

```
10 ERG = 1
20 INPUT "FAKULTAET";F
30 FOR I = 1 TO F
40 ERG = ERG * F
50 NEXT I
60 PRINT F;" FAKULTAET BETRAEGT ";ERG
70 END
```

Dieses Programm berechnet die Fakultät einer eingegebenen Zahl. Hierbei ist zu beachten, daß nur Fakultäten von positiven, ganzen Zahlen zulässig sind. Fakultäten von Zahlen, die größer als 33 sind können nicht mit dem COMMODORE 64 berechnet werden, da das Ergebnis seinen Zahlenbereich überschreiten würde.

Farbe

Der COMMODORE 64 bietet die Möglichkeit 16 Farben auf dem Bildschirm zu erzeugen. Dies kann nun auf zwei Arten geschehen und zwar einmal über die Tastatur und einmal durch entsprechende *CHR$ Befehlen. Die Farberzeugung über die Tastatur geschieht mit Hilfe der *CTRL-Taste und der *COMMODORE-Taste. Die Tasten zur Farberzeugung sind die Zifferntasten 1-8 in Zusammenhang mit einer der vorher genannten Tasten.

Ziffer	CTRL	COMMODORE
1	schwarz	orange
2	weiß	braun
3	rot	hellrot
4	türkis	grau 1
5	violett	grau 2
6	grün	hellgrün
7	blau	hellblau
8	gelb	grau 3

Diese Art der Farbausgabe kann durch ein *PRINT Statement erreicht werden.

Beispiel :

```
10 PRINT "CTRL3COMMODORE 64"
20 GOTO 10
```

In diesem Beispiel wird COMMODORE 64 in rot auf dem Bildschirm ausgegeben.
Die andere Art Farbe auf den Bildschirm zu bringen ist sie direkt über den entsprechenden CHR-Code anzusprechen durch ein *CHR$ Statement.

Beispiel :

```
10 PRINT CHR$(28);"COMMODORE 64"
20 GOTO 10
```

Auch in diesem Beispiel wird das Wort "COMMODORE 64" in rot auf dem Bildschirm ausgegeben.

Die Farben haben folgende CHR-CODES :

weiß	5
rot	28
grün	30
blau	31
schwarz	144
violett	156
gelb	158
türkis	159
orange	129
braun	148
hellrot	150
grau 1	151
grau 2	152
hellgrün	153
hellblau	154
grau 3	155

Auf die oben beschriebene Art und Weise können nun Zeichen farbig auf dem Bildschirm wiedergegeben werden. Der Hintergrund bleibt aber, wie gehabt, blau. Es gibt nun aber noch die Möglichkeit, mit Hilfe von *POKE-Statements den Hintergrund und den Rahmen einzufärben.

Beispiel :

```
POKE 53281,5
```

Durch diesen Befehl wird der Hintergrund grün gefärbt. Auf die gleiche Weise kann auch der Rahmen eingefärbt werden.

Beispiel :

```
POKE 53280,2
```

Nun wurde der Bildschirmrahmen rot gefärbt. Wenn durch *POKE-Befehle die Farben geändert werden sollen, so sind ihnen folgende Zahlenwerte zugeordnet :

Farbe	Zahlenwert
schwarz	0
weiß	1
rot	2
türkis	3
violett	4
grün	5
blau	6
gelb	7
orange	8
braun	9
hellrot	10
grau 1	11
grau 2	12
hellgrün	13
hellblau	14
grau 3	15

Farbspeicher

Äquivalent zum *Bildschirmspeicher. Hier wird der Farbcode des Zeichens hinterlegt, das durch Manipulation des Bildschirmspeichers auf eine bestimmte Position des Bildschirms gebracht worden ist. Der Farbspeicher löst den Bildschirm in 24 Zeilen zu 40 Spalten auf. Jedes Feld kann nun direkt mit einer der 16 Farben belegt werden.

Beispiel :

POKE 55296,3

Hierdurch wird die linke obere Ecke des Bildschirms in türkis gefärbt. Soll nun ein "*" in der linken oberen Ecke des Bildschirms türkis erscheinen, so sind folgende Befehle nötig:

POKE 1024,42
POKE 55296,3

Die Adressen des Farbspeichers liegen zwischen 55296 und 56295. Sie können folgendermaßen berechnet werden :

Adresse = 55296 + Spalte + 40 * Zeile

Hierbei ist zu beachten, daß die erste Zeile und die erste Spalte jeweils den Wert Null haben und dem zur Folge zum Beispiel die 40. Spalte den Wert 39 hat.

Fehlermeldungen des COMMODORE 64

Treten in einem Programm Fehler auf, oder gibt es Schwierigkeiten beim Laden eines Programms, so meldet das der COMMODORE 64, indem er eine Fehlermmeldung auf den Bildschirm bringt. Im Folgenden wird eine Auflistung der Fehlermeldungen gegeben, sowie eine Erklärung ihrer jeweils zu Grunde liegenden Ursachen :

BAD DATA ERROR

Dieser Fehler tritt auf, wenn von einem *File (Datei) Buchstaben oder Sonderzeichen gelesen wurden, das Programm aber numerische Daten erwartet. Behoben werden kann dieser Fehler dadurch, daß die Variable, in die die Daten eingelesen werden sollen, als *STRING-Variable gekennzeichnet wird.

BAD SUBSCRIPT ERROR

Versucht man einem *Array Daten auf ein Feld zuzuweisen, das außerhalb des im *DIM-Statement festgelegten Bereichs liegt, so tritt dieser Fehler auf. Dadurch daß der Bereich im *DIM-Statement vergrößert wird, kann dieser Fehler behoben werden.

CAN'T CONTINUE ERROR

Wurde das Programm durch Betätigung der *RUN/STOP Taste unterbrochen oder kam es durch das Auftreten eines Fehlers zu einer Programmunterbrechung und wurden Zeilen verändert, so kann das *CONT Statement nicht ausgeführt werden, das Programm muß mit *RUN neu gestartet werden.

DEVICE NOT PRESENT ERROR

Wurde durch eines der Ein-/Ausgabestatements ein Gerät angesprochen, das nicht angeschlossen oder eingeschaltet war, so tritt dieser Fehler auf. Behoben werden kann dieser Fehler dadurch, daß das entsprechende Gerät angeschlossen oder eingeschaltet wird. Die Ein-/Ausgabestatements bei denen dieser Fehler auftreten kann sind : OPEN, CLOSE, CMD, PRINT#, INPUT#, GET#.

DIVISION BY ZERO ERROR

Die Division durch Null ist mathematisch nicht definiert und kann daher vom COMMODORE 64 nicht durchgeführt werden. Tritt dieser Fehler auf, so sollten die Variablen, mit denen in dem entsprechenden Statement gerechnet wird, daraufhin untersucht werden, ob sie den Wert Null enthalten oder nicht. Wenn sichergestellt ist, daß die Variablen, durch die dividiert wird, nicht gleich Null sind, so tritt dieser Fehler nicht mehr auf.

EXTRA IGNORED

Wurde versucht einem *INPUT Statement mehr Daten zuzuweisen, als vom Programm aus vorgesehen, so kommt es zu dieser Fehlermeldung. Eine häufige Ursache dieses Fehlers ist, wenn numerische Daten durch *INPUT erfasst werden sollen und statt des Dezimalpunktes ein Komma eingegeben wird. In diesem Fall versucht der COMMODORE 64 die Daten auf zwei Variablen zuzuweisen. Bei Auftreten dieses Fehlers wird das Programm

nicht unterbrochen, es werden nur die Daten gelesen, die links vom Komma stehen.

FILE NOT FOUND

Zu diesem Fehler kommt es, wenn ein Programm oder eine Datei geladen werden soll und sich diese nicht auf dem angesprochenen Datenträger befindet. Die Ursache für diese Fehlermeldung liegt zum Beispiel darin, daß das benötigte Programm sich nicht auf der Diskette befindet oder der Programmname falsch geschrieben wurde.

FILE NOT OPEN ERROR

Wurde eine Datei mit CMD, PRINT#, INPUT# oder GET# angesprochen und wurde diese Datei nicht vorher mit OPEN geöffnet, so erscheint diese Fehlermeldung. Dieser Fehler kann behoben werden, indem man einen OPEN-Befehl für die entsprechende Datei einfügt.

FILE OPEN ERROR

Es wurde versucht eine Datei zu öffnen, die bereits geöffnet war. Die Ursache dieses Fehlers kann einmal sein, daß versehentlich eine Filenummer zweimal verwendet wurde, dann ist der Fehler dadurch zu beheben, daß diese Filenummer verbessert wird. Zum anderen kann dieser Fehler aber auch dadurch auftreten, daß ein File wirklich zweimal geöffnet wurde, dann muß nur das zweite OPEN Statement gelöscht werden.

FORMULA TOO COMPLEX

Zwei Bedingungen können diese Fehlermeldung verursachen. 1. es wurde versucht einen String zu lesen, der zu lang ist, in diesem Fall sollte versucht werden den String in mindestens zwei Teile zu zerlegen. Die 2. Fehlerursache kann sein, daß ein mathematischer Ausdruck zuviele Klammern enthält, in

diesem Fall sollte der Ausdruck noch einmal überarbeitet und neu geschrieben werden.

ILLEGAL DIRECT ERROR

Tritt diese Fehlermeldung auf, so wurde versucht im *COMMAND Modus ein INPUT Statement zu benutzen.

ILLEGAL QUANTITY ERROR

Eine Zahl, die als Argument einer Funktion oder einer Anweisung verwendet wurde liegt außerhalb des zulässigen Bereichs. Es kann zum Beispiel versucht worden sein den SINUS einer negativen Zahl zu berechnen. Kommt es zu dieser Fehlermeldung, so sollte die entsprechende Variable daraufhin untersucht werden, ob sie einen unzulässigen Wert enthält.

LOAD ERROR

In diesem Fall gibt es Probleme beim Laden von der DATASETTE oder dem Diskettenlaufwerk. Die Ursache dieser Probleme kann sehr vielschichtig sein, einmal ist es möglich, daß versucht wurde Daten einzulesen, die der COMMODORE 64 nicht lesen kann, dann ist es möglich, daß der *Datenträger (Casette oder Diskette) defekt ist, oder daß gar eines der Geräte defekt ist. In jedem Fall sollte mehrmals der Ladevorgang wiederholt werden, um ganz sicher zu gehen, daß wirklich kein Bedienerfehler vorliegt.

NEXT WITHOUT FOR ERROR

Auch hier sind mehrere Fehlerursachen denkbar. 1. es wurden mehrere FOR - NEXT Schleifen nicht korrekt verschachtelt oder 2. die im NEXT Statement angegebene Variable stimmt nicht mit der im FOR Statement überein. Um diesen Fehler zu beheben ist es ratsam, den entsprechenden Programmteil noch einmal genau zu überprüfen.

NOT INPUT FILE ERROR

Es wurde versucht mit INPUT# oder GET# Daten von einem *File zu lesen, der nicht als Eingabefile erklärt wurde. In diesem Fall sollte die entsprechende Programmstelle noch einmal genau untersucht werden.

NOT OUTPUT FILE

Wurde versucht durch PRINT# *Daten an einen *File zu senden, der nur als Eingabefile geöffnet war, so kommt es zu dieser Fehlermeldung. Der Programmteil, in dem dieser Fehler auftritt sollte einer genauen Untersuchung unterzogen werden.

OUT OFF DATA ERROR

Wenn dieser Fehler auftritt, wurde versucht durch ein *READ Statement *DATA Zeilen zu lesen, die nicht vorhanden sind. Behoben werden kann dieser Fehler indem DATA Zeilen angefügt werden, oder in dem das letzte READ Statement gestrichen wird.

OUT OF MEMORY ERROR

In diesem Fall steht kein Speicherplatz im *RAM-Bereich mehr für Daten oder Programmtext zur Verfügung. Auftreten kann dieser Fehler wenn zu viele *FOR-NEXT Schleifen verschachtelt wurden, oder wenn zu viele *Unterprogramme ineinander verschachtelt worden sind. In einem dieser Fälle bleibt nur die Möglichkeit das Programm zu überarbeiten, um den Fehler zu beseitigen.
Dieser Fehler kann aber auch noch auftreten, wenn das Wort *READY mit *RETURN quitiert wurde. In diesem Fall braucht die Meldung nicht ernst genommen zu werden. Es kann normal weitergearbeitet werden.

OVERFLOW ERROR

Ist in einem Programmm eine Zahl berechnet worden, die über den Zahlenbereich des COMMODORE 64 hinausgeht, so tritt diese Fehlermeldung auf. Der Zahlenbereich des COMMODORE 64 läßt nur Zahlen bis 1.70141183 * 10^38 zu. Behoben werden kann dieser Fehler dadurch, daß sichergestellt wird, daß solch große Zahlen nicht auftreten können.

REDIM'D ARRAY ERROR

In einem BASIC Programm darf ein *Array nur einmal dimensioniert werden. Die Dimensionierung kann auf zwei Arten erfolgen und zwar 1. wird eine Element eines Arrays zum ersten Mal in einem Programm angesprochen, so wird der Array automatisch auf 10 Felder dimensioniert, jedes *DIM Statement im Programmablauf, das sich auf diesen Array bezieht führt dann zu dieser Fehlermeldung. 2. ein Array kann auch explizit durch das DIM Statement dimensioniert werden, jedes weitere DIM im Programmablauf führt dann zu diesem Fehler. Behoben werden kann er dadurch, daß das überflüssige DIM entfernt wird.

REDO FROM START

Wurde eine INPUT Variable als numerisch erklärt und wird nun versucht auf diese Variable einen *String zuzuweisen, so kommt es zu diesem Fehler. Der COMMODORE 64 führt dann das INPUT Statement noch mal aus, und zwar so lange bis ein numerischer Wert eingegeben wurde.

RETURN WITHOUT GOSUB

Im Programmablauf wurde ein *RETURN erreicht, ohne daß ein *GOSUB vorangegengen ist. Eine häufige Ursache dieses Fehlers ist, daß das *END, als letztes Statement des Hauptprogramms vergessen worden ist, oder daß in ein Unterprogramm rein gesprungen wurde durch GOTO. Durch Beseitigung dieser

Fehlerursachen kann der Fehler behoben werden.

STRING TOO LONG

Der COMMODORE 64 kann nur Strings verarbeiten, die maximal 255 Zeichen lang sind. Wurde versucht mit einem String zu arbeiten, der länger ist, so kommt es zu diesem Fehler.

SYNTAX ERROR

Der wohl häufigste Fehler, dieser Fehler kann eine sehr große Zahl von Ursachen haben, meist liegt ihm ein Schreibfehler zu Grunde, es wurde zum Beispiel eine Klammer vergessen, oder eine Klammer zu viel gesetzt oder es wurde ein BASIC Schlüsselwort falsch geschrieben.

TYPE MISMATCH ERROR

Dieser Fehler tritt auf, wenn bei einer Rechenoperation eine *Stringvariable benutzt wurde oder bei einer Stringoperation eine numerische Variable. Um den Fehler zu beheben sollten die Variablen noch einmal genau untersucht werden.

UNDEF'D FUNCTION

Es wurde auf eine selbstdefinierte Funktion Bezug genommen, die nicht definiert worden ist, bzw. deren Definitionsstatement noch nicht durchlaufen worden ist.

UNDEF'D STATEMENT ERROR

Es wurde versucht eine Programmzeile durch GOTO, GOSUB oder RUN anzusprechen, die noch nicht existiert. Zu beheben ist der Fehler durch Uberprüfung der Sprungadresse bzw. durch Erstellen der Programmzeile, die angesprungen werden soll.

VERIFY ERROR

Nachdem ein Programm auf *Diskette oder *Cassette abgespeichert wurde und mit dem *VERIFY Befehl die Übereinstimmung zwischen Speicherinhalt und Band bzw. Disketteninhalt überprüft wurde, ist hier eine Diskrepanz festgestellt worden. In diesem Fall ist es ratsam das Programm noch einmal auf einer anderen Diskette oder Cassette abzuspeichern. Tritt dieser Fehler häufiger auf, so ist es möglich, daß das *Diskettenlaufwerk bzw. die *DATASETTE nachjustiert werden müssen.

Fehlermeldungen des Diskettenlaufwerks

Beim Arbeiten mit dem Diskettenlaufwerk kann es zu Fehlern kommen, in diesem Fall werden die Fehler nicht vom COMMODORE 64 sondern vom *DOS des Diskettenlaufwerks gemeldet. Das *DOS gibt seine Fehlermeldungen nicht nur als Text aus, sondern auch über den Fehlerkanal als Zahlencode. Dieser Zahlencode kann durch folgende Befehlssequenz gelesen werden.

```
Open 15,8,1
INPUT#15 A,B$,C
PRINT A;B$;C
```

Die Zahlencodes spezifizieren den angezeigten Fehler näher. Im folgenden werden diese Fehlermeldungen aufgelistet und erläutert.

READ ERROR

Dieser Fehler tritt immer dann auf, wenn das Diskettenlaufwerk nicht in der Lage ist ein Programm oder Daten korrekt von der Diskette zu lesen.

Fehlercode 20
Das Diskettenlaufwerk kann den Anfang eines Blocks nicht finden, da ein Track oder ein Sector defekt ist.

Fehlercode 21
Es wurde versucht eine nicht formatierte Diskette zu lesen, eine weitere Möglichkeit ist, daß ein Hardwarefehler vorliegt.

Fehlercode 22
Es wurde versucht über einen Direktzugriffsbefehl einen Block zu lesen, der nicht existiert.

Fehlercode 23
Bei der Ausführung des *VERIFY-Befehls wurde ein Fehler entdeckt. Dieser Fehler hat sehr oft seine Ursache in einer Hardwarestörung oder darin, daß die Diskette defekt ist.

Fehlercode 24
Resultiert fast immer aus einem Hardwarefehler.

Fehlercode 27
Tritt auf bei Ausführung des *VERIFY-Befehls auf, wenn ein Fehler im Programmheader entdeckt wird. Eine weitere Ursache dieser Fehlermeldung kann eine Hardwarestörung sein.

WRITE ERROR

Ist es zu diesem Fehler gekommen, so liegt immer eine Diskrepanz zwischen den Daten die auf die Diskette geschrieben wurden und den Daten die eigentlich auf die Diskette geschrieben werden sollten.

Fehlercode 25
Wenn die Daten, die auf die Diskette geschrieben werden nicht mit den Daten übereinstimmen, die aus dem Disk-RAM gelesen wurden, tritt dieser Fehler auf.

Fehlercode 28

Ist die Diskette nicht formatiert oder defekt oder liegt ein Hardwarefehler vor so kommt es zu diesem Fehler.

WRITE PROTECTION ON

Die Ursache dieses Fehlers ist, daß versucht wurde auf eine schreibgeschützte Diskette zu schreiben.

DISK ID MISMATCH

Wird ein Direktzugriffsbefehl zum Arbeiten mit einer unformatierten Diskette oder einer defekte Diskette verwendet, so bringt das DOS diese Fehlermeldung auf den Bildschirm.

SYNTAX ERROR

Wird ein Befehl nicht richtig geschrieben oder ist der Befehl zu lang , so daß das Diskettenlaufwerk einen Befehl nicht verstehen kann, so ist das Resultat diese Fehlermeldung.

RECORD NOT PRESENT

Dieser Fehler tritt auf, wenn versucht wird einen Satz zu lesen, der nicht mehr auf der Diskette steht.

OVERFLOW IN RECORD PRINT#

Wenn versucht wird mehr als eine definierte Anzahl von Zeichen durch ein PRINT# Statement auf die Diskette zu schreiben, so erscheint diese Fehlermeldung.

FILE TOO LARGE

Die Datei, die abgespeichert werden soll ist so groß, daß sie nicht mehr auf die Diskette paßt.

WRITE FILE OPEN

Der File ist bereits geöffnet worden bzw. er ist bisher nicht geschlossen worden. Diese Meldung bezieht sich nur auf Files, die auf die Diskette schreiben sollen.

FILE NOT OPEN

Es wurde versucht auf einen File zuzugreifen, der noch nicht geöffnet worden ist.

FILE NOT FOUND

Dieser Fehler tritt auf, wenn versucht wird, eine Datei oder ein Programm zu laden, das entweder nicht existiert oder sich nicht auf der eingelegten Diskette befindet. Eine häufige Ursache dieses Fehlers ist, daß der Filename nicht richtig geschrieben wurde.

FILE EXISTS

Diese Fehlermeldung wird vom Diskettenlaufwerk augegeben, wenn ein File auf die Diskette geschrieben werden soll, dessen Name schon existiert.

FILE TYPE MISMATCH

Wird versucht einen File mit einem anderen Typ zu verarbeiten, als er auf der Diskette gespeichert ist.

NO BLOCK

Soll ein Block durch den BLOCK-ALLOCATE Befehl als belegt gekennzeichnet werden, der im Moment benutzt wird, so kommt es zu dieser Fehlermeldung.

ILLEGAL TRACK AND SECTOR

Diese Fehlermeldung wird von dem Diskettenlaufwerk ausgegeben, wenn auf einen Track oder einen Sector zugegriffen wird, der nicht verfügbar ist oder nicht existiert.

ILLEGAL SYSTEM T OR S

Wird auf einen Track oder einen Sector zugegriffen, auf den nicht zugegeriffen werden kann, so kommt es zu dieser Fehlermeldung.

NO CHANNEL

Wenn mehr als die maximal mögliche Anzahl von Kanälen eröffnet wird, so wird diese Fehlermeldung auf dem Bildschirm ausgegeben. Es ist möglich bis zu 5 sequentielle Dateien oder bis zu 6 Direktzugriffskanäle gleichzeitig geöffnet zu haben.

DIRECTORY CHANNEL

Bestehen irgendwelche Probleme mit der *BAM, so kommt es zu diesem Fehler. Behoben werden kann er oft durch eine neue *Disketteninitialisierung.

DISK FULL

Wird versucht mehr als 144 Dateien auf einer Diskette anzulegen, so ist die Bedingung für diese Fehlermeldung erfüllt.

Fehlermeldungen, deutsch

Der COMMODORE 64 bringt seine Fehlermeldungen gewöhnlich in englischer Sprache, es ist aber möglich durch ein Programm

diese Fehlermeldungen auf deutsch zu bringen. Dazu ist ein kleines Programm nötig, das in Maschinensprache geschrieben ist. Dieses Programm ist in dem Buch "64 Tips und Tricks" zu finden".

Feld

1. Synomymer Ausdruck für *Array.
2. Einzelnes Element eines *Arrays.

Fenster

Bestimmter Teil eines Bildschirms, in dem ausgewählte Informationen dargestellt werden können. Es ist zum Beispiel möglich die Information welcher Track von der Diskette gelesen wird in einem festgelegten Teil des Bildschirms wiederzugeben.

Festplatte

Externer Massenspeicher eines Computers. Zu vergleichen ist eine Festplatte mit einer *Diskette, die Unterschiede zwischen diesen Speichermedien sind :

1. Die Festplatte hat einen größeren Durchmesser als die Diskette.
2. Die Festplatte hat eine höhere Speicherkapazität als die Diskette.
3. Die Disketten können beliebig ausgewechselt werden, die Festplatte hingegen nicht.
4. Für *Homecomputer gibt es keine Festplattenlaufwerke.

Eine Festplatte besteht, genau wie eine Diskette, aus einer

Kuststoffscheibe, deren Oberfläche magnetisiert werden kann.

Festwertspeicher

Darunter wird ein Speicher verstanden, der sich innerhalb des Computers befindet und bei Abschalten des Stroms nicht gelöscht wird. Bei vielen Computern kann dieser Festwertspeicher nicht verändert, sondern nur gelesen werden. Der Festwertspeicher wird auch als *ROM bezeichnet, er beinhaltet wichtige Teile des *Betriebssystems.

Fetch

Abruf von Daten von einem Peripheriegerät.

FIELD

Anweisung des MIKROSOFT-BASIC. Diese Anweisung dient dem Direkktzugriff auf Dateien. Sie teilt den Datenpuffer in einzelne Felder auf, die separate Namen bekommen und deren Länge genau festgelegt werden kann.

SYNTAX :

FIELD # Dateinummer, Größe, AS Stringvariable,Größe,AS Stringvariable.....

Diesen Befehl gibt es im BASIC des COMMODORE 64 nicht, er kann auch nur unter sehr großem Aufwand simuliert werden.

File

Synonymer Ausdruck für *Datei.

File management

Dateiverwaltung oder Datehandhabung, hierbei geht es zum Beispiel darum, wie eine Datei aufgebaut ist.

File protection

Dateischutz

FILES

Befehl des MIKROSOFT-BASIC. Die Aufgabe dieses Befehls ist es die Name der auf der Diskette gepseicherten Dateien anzuzeigen.

SYNTAX :

FILES Auswahl

Ein Teil des Umfanges dieses Befehls kann auch mit dem BASIC des COMMODORE 64 realisiert werden, es ist möglich alle Namen, der auf der Diskette gespeicherten Dateien, auszugeben und zwar mit dem Befehl

LOAD "$",8

Der MIKROSOFT Befehl FILES hat noch mehrere Varianten, die im COMMODORE 64 BASIC nicht realisiert werden können.

Fill character

Füllzeichen, muß ein *Datensatz eine bestimmte Länge aufweisen, reicht sein Inhalt aber nicht aus, so muß der Rest mit Füllzeichen aufgefüllt werden.

Firmware

Darunter wird *Software verstanden, die einem bestimmten Computer fest zugeordnet ist und nicht mehr verändert werden kann. Dies sind zum Beispiel Betriebssysteme, die nicht als *Festwertspeicher vorliegen, sondern erst geladen werden müssen. Diese spezielle Software wird von den Anwendern meist wie *Hardware gehandhabt.

FIX

Funktion des MIKROSOFT-BASIC. Seine Aufgabe ist die Ermittlung des ganzzahligen Anteils einer Zahl, ähnlich dem *INT-Befehl.

SYNTAX :

FIX (Argument)

Im MIKROSOFT-BASIC ist diese Funktion nötig, da der *INT-Befehl nur für positive Zahlen verwendbar ist. Im BASIC des COMMODORE 64 hingegen ist dies nicht nötig, da hier der *INT-Befehl auch den ganzzahligen Anteil negativer Zahlen berechnet.

Fixed disk

*Festplatte, Magnetplatte, die nicht gewechselt werden kann.

Flag

Kennzeichen. Flags werden in der *Maschinensprache benötigt, um festzustellen, ob eine Bedingung erfüllt ist.

Floppy-Disk

Synonymer Ausdruck für *Diskette.

Floppy-Disk Controller

Teil des Diskettenlaufwerks, der den Betrieb der Diskette kontrolliert.

Floppy-Disk Drive

Darunter wird das *Diskettenlaufwerk verstanden.

Flow chart

Unter einer Flow Chart wird ein *Programmablaufplan

verstanden. Bevor mit der Programmerstellung begonnen wird sollte ein Programmablaufplan erstellt werden.

Foreground program

Programm, das im Vordergrund läuft, während im Hintergrund zum Beispiel gedruckt wird.

FOR...NEXT

Anweisung, die es in fast allen BASIC-Dialekten gibt. Diese Anweisung bietet die Möglichkeit eine Anzahl von Statements so oft wie gewünscht zu durchlaufen.

SYNTAX :

```
FOR Variable = Zahl TO Wert
...
...
NEXT Variable
```

Sollen zum Beispiel die Quadratwurzeln der Zahlen von 1 bis 100 berechnet werden so läßt sich dieses Vorhaben mit Hilfe einer FOR...NEXT Schleife leicht realisieren, wie das folgende Beispiel zeigt :

Beispiel :

```
10 FOR I = 1 TO 100
20 PRINT I,SQR(I)
30 NEXT I
```

Genau so leicht ist es die Quadratwurzel der Zahlen von 250 bis 2836 zu berechnen, es muß in dem Beispiel nur die "1"

durch "250" und die "100" durch "2836" werden. Ein sinnvoller Anwendungsbereich der FOR...NEXT Schleifen ist die Arbeit mit *Arrays, die einzelnen Elemente eines Arrays können in dieser Form einfach bearbeitet werden. Zum Beispiel ist es möglich eine festgelegte Anzahl von Namen durch eine FOR...NEXT Schleife in einem Array zu erfassen.

Beispiel :

```
10 DIM A$(15)
20 FOR I = 1 TO 15
30 INPUT A$(I)
40 NEXT I
```

Genauso einfach ist es auch möglich mehrdimensionale Arrays zu bearbeiten.

FOR...NEXT, geschachteltes

Die FOR...NEXT-Schleifen haben die angenehme Eigenschaft sich ineinander verschachteln zu lassen, das heißt, innerhalb einer FOR...NEXT-Schleife wird eine andere angelegt.

Beispiel :

```
10 FOR I = 1 TO 25
20 FOR K = 1 TO 10
30 PRINT K;" * ";I;" = ";I*K
40 NEXT K
50 PRINT
60 NEXT I
```

In diesem Beispiel wird das "1 * 1" der Zahlen von 1 bis 25. Bei der Verschachtelung von FOR...NEXT-Schleifen ist darauf zu achten, daß sie in der umgekehrten Reihenfolge abgeschlossen werden, in der sie begonnen wurden. Ist der

erste Zähler einer Reihe von FOR...NEXT-Schleife ein I, so muß das letzte Statement NEXT I sein. Zwischen zwei FOR-Statements können auch beliebige andere Statments stehen.

Beispiel :

```
10 FOR I = 1 TO 25
20 PRINT I;"er Einmaleins"
30 FOR K = 1 TO 10
40 PRINT K;" * ";I;" = ";I*K
50 NEXT K
60 PRINT
70 NEXT I
```

FORTH

Eine der jüngsten *höheren Programmiersprachen, bei der Entwicklung wurde versucht die Vorteile der anderen höheren Programmiersprachen zu berücksichtigen, ohne deren Nachteile zu übernehmen. FORTH hat folgende besonderen Vorteile:

1. Die Programme benötigen nur einen geringen Speicherplatz.
2. FORTH benötigt nur sehr wenige Ein-/Ausgabeleitungen.
3. Aus dem Zweiten geht hervor, daß FORTH theoretisch ohne ein Diskettenlaufwerk auskommen kann.

Der größte Vorteil von FORTH ist, daß es für den COMMODORE 64 verfügbar ist und zwar als Steckmodul und auf Diskette. Außerdem ist zu bemerken, daß FORTH die Möglichkeit bietet eigene Befehle zu generieren, so daß man sich quasi eine eigene Sprache erzeugen kann.

FORTRAN

Höhere Programmiersprache, die ihren hauptsächlichen Anwendungsbereich bei der Lösung technisch-mathematischer Probleme hat. FORTRAN ist die Abkürzung für "Formula Translation", diese Sprache wurde bereits Ende der fünfziger Jahre von IBM entwickelt worden. FORTRAN ist im Laufe der Zeit immer weiter entwickelt worden, die heute meist benutzte Version ist FORTRAN IV, das 1966 standardtisiert wurde. Bisher gibt es noch keine FORTRAN-Version für *Homecomputer, wohl aber für *Mikrocomputer.

FRE

Funktion des BASIC des COMMODORE 64. Sie gibt die Zahl der freien Bytes im Speicher an.

SYNTAX :

FRE(0)

Hierbei ist zu beachten, daß die ausgegebene Zahl negativ ist. Wird zu dieser Zahl 65536 addiert, so ist das Ergebnis die Zahl der tatsächlichen freien Bytes im Speicher wiedergegebben. Auf dem Bildschirm wird diese Zahl durch folgendes Statement ausgegeben :

PRINT FRE(0)

Frequenzumtastung

Darunter versteht man ein Verfahren um Daten auf einer Cassette aufzuzeichnen. Die Grundlage dieses Verfahrens ist, daß der binäre Code, also die Nullen und Einsen, durch zwei

verschiedene Frequenzen dargestellt werden. Dieses Verfahren wird auch vom COMMODORE 64 und der *Datasette verwendet. Wird eine Cassette, die Programme enthält in einen normalen Audiocassettenrecorder eingelegt und abgespielt, so ist diese Frequenzumtastung deutlich zu hören.

Füllsignal

Synonymer Ausdruck für Dummy. Diese Signale werden nur aus formalen Gründen verarbeitet, daß heißt eine Datei, die eine bestimmte Anzahl von Sätzen enthalten soll, wird , falls sie weniger Sätze enthält, durch Leerzeichen aufgefüllt.

Function key

Funktionstaste, der COMMODORE 64 hat 8 Funktionstaste, *F1 - F8, diese Tasten können mit eigene Funktionen belegt werden.

Funktion

Mathematisch betrachtet versteht man darunter eine veränderliche Größe, die von einer anderen veränderlichen Größe abhängt. Ein Beispiel für eine Funktion ist :
X = (2 * A + 4 * B) / A + B

In diesem Beispiel hängt die veränderliche Größe X von der ebenfalls veränderlichen Größe (2 * A + 4 * B) / A + B ab.

Funktion, Definition einer

Der COMMODORE 64 bietet dem Benutzer die Möglichkeit eigene Funktionen zu definieren. Dies geschieht mit Hilfe des *DEF FN Statements.

Funktionstaste

Darunter versteht man eine Taste, die vom Programmierer selbst belegt werden kann. Der COMMODORE 64 hat 8 Funktionstasten, die mit *F1 - F8 bezeichnet sind.

G

Taste der Tastatur des COMMODORE 64 mit dem *Tastaturcode 26. Durch diese Taste können die Zeichen G, □ g, ◫ erzeugt werden. Die *ASCII-Codes dieser Zeichen lauten :

	Großschrift/Graphik	Groß-/Kleinschrift
G	71	199
g	-	71
□	165	165
◫	103	-

Diese Zeichen haben die folgenden *Bildschirmcodes:

	Großschrift/Grapik	Groß-/Kleinschrift
G	7	71
g	-	7
□	101	101
◫	71	-

Die Umschaltung zwischen den Modi Großschrift/Graphik und den Groß-/Kleinschrift kann durch gleichzeitige Betätigung der Tasten *SHIFT und COMMODORE vorgenommen werden.

Ganzzahl

Dies ist eine Zahl ohne Nachkommastellen, also zum Beispiel 5, 8, -3 usw. Der COMMODORE 64 läßt es zu, den ganzzahligen Anteil einer Zahl zu ermitteln und zwar durch den Befehl *INT.

Gap

Kluft, Spalte, Lücke

Garbage collection

Wird mit dem COMMODORE 64 durch ein BASIC-Programm ein großer Text verarbeitet, so löscht der Computer die nicht mehr benötigten Teile nicht aus seinem Speicher, sondern er setzt nur den Pointer weiter, dadurch ist der verfügbare Speicherplatz nach einer gewissen Zeit voll und muß wieder verfügbar gemacht werden. Dieser Vorgang des Verfügbarmachens wird "Garbage collection" genannt. Wörtlich übersetzt heißt Garbage collection Müll sammeln.

Garbage in - Garbage out

Ausdruck von Programmierern der besagt, daß ein Computer nur das tut, was ihm gesagt wird. Wird also etwas logisch Falsches eingegeben, so kann auch nur etwas Falsches vom Computer ausgegeben werden.

Genauigkeitsschranke

Wird ein Wert näherungsweise berechnet, so muß festgelegt werden, wann der Wert genau genug berechnet ist, dafür wird eine Genauigkeitsschranke definiert. Dargestellt wird sie meist als Differenz zwischen oberem und unterem Näherungswert.

Beispiel :

Berechnung der Kubikwurzel einer Zahl.

```
10 INPUT "ZAHL                    : ";Z
20 INPUT "OBERGRENZE              : ";O
30 INPUT "UNTERGRENZE             : ";U
40 INPUT "GENAUIGKEITSSCHRANKE : ";G
50 INPUT "SCHRITTWEITE            : ";S
60 OW = O * O * O
70 UW = U * U * U
80 IF OW - UW < G THEN 120
90 O = O - S
100 U = U + S
110 GOTO 60
120 ERG = (OW + UW)/2
130 PRINT "DIE KUBIKWURZEL VON ";Z;" BETRÄGT ";ERG
140 END
```

Die Genauigkeitsschranke kommt in der Zeile 80 zum Tragen, hier wird geprüft, ob die Genauigkeit des Ergebnisses hoch genug ist.

Geräusche

Der COMMODORE 64 ist ein vollwertiger Synthesizer, das heißt es ist möglich mit ihm alle Geräusche vom Peitschenknall bis zum Türklappen zu erzeugen, genauso wie es möglich ist mit dem COMMODORE 64 *Musik zu machen. Verantwortlich für den Sound ist der SID 6581, ein Peripheriebaustein der 6500-Familie. Er verfügt über :

1. 3 einzeln programmierbare Stimmen
2. 4 mischbare Schwingungsarten für jede Stimme
3. 3 mischbare Filter (Hoch,Tief und Bass) ebenfalls für jede Stimme
4. einen Hüllkurvengenerator für jede Stimme
5. 2 kaskandierbare Ringmodulatoren für jede Stimme

6. Verfremdungsmöglichkeiten für externe Signalquellen
7. 2 8-Bit A-D Wandler.

Eine ausführliche Beschreibung des SID 6581 findet sich in dem Buch "64-intern".

GET

Befehl, den es in fast allen BASIC-Versionen gibt. Seine Aufgabe ist es ein Zeichen von der Tastatur einzulesen und dieses Zeichen einer Variablen zuzuweisen.

SYNTAX :

GET Variablenname

Die Variable, der das eingelesene Zeichen zugewiesen wird kann sowohl numerisch, als auch alphanumerisch sein, meist werden alphanumerische Variablen verwendet, da es zu einer Fehlermeldung kommt, wenn ein Buchstabe oder ein Sonderzeichen auf eine numerische Variable zugewiesen wird.

Beispiel :

```
10 GET A$
20 IF A$ = " " THEN 10
30 IF A$ = "J" THEN ...
40 IF A$ = "N" THEN END
50 GOTO 10
```

In diesem Beispiel wird die Tastatur daraufhin abgefragt, ob ein J oder ein N eingegeben wurde. Diese Technik ist zum Beispiel dann nützlich, wenn in einem Programm abgefragt werden soll, ob der Benutzer das Programm beenden will oder nicht.

Get#

Befehl, der in praktisch allen BASIC-Versionen existiert, er wird benötigt, um von einem Datenfile ein Zeichen einzulesen.

SYNTAX :

GET#Kanalnummer,Variablenname

Soll ein Datenfile zum Beispiel von der *DATASETTE eingelesen werden, so ist dieser Befehl sehr nützlich.

GOSUB

Befehl, der in allen BASIC-Dialekten vorkommt. Durch diesen Befehl wird ein *Unterprogramm aufgerufen.

SYNTAX :

GOSUB Zeilennummer

Der Computer fährt mit der Zeilennummer fort, die in dem GOSUB-Befehl steht. Ein *Unterprogramm muß durch den Befehl *RETURN abgeschlossen werden. Findet der Computer den Befehl RETURN so setzt er das Programm mit dem Statement fort, das dem Befehl GOSUB folgt. Der Befehl GOSUB dient also dazu ein Unterprogramm aufzurufen und der Befehl RETURN dazu nach Beendigung des Unterprogramms in das Hauptprogramm zurückzukehren.

Beispiel :

```
10 REM HAUPTPROGRAMM
20...
30...
40 GOSUB 100
...
...
100 REM UNTERPROGRAMM
110...
120..
130 RETURN
```

In diesem Beispiel wird ein Unterprogramm, das in der Zeile 100 beginnt in Zeile 40 aufgerufen. Nachdem das Unterprogramm abgearbeitet worden ist und die Zeile 130 erreicht wurde, wird mit dem Hauptprogramm weitergearbeitet. Es ist sicherzustellen, daß kein Unterprogramm bearbeitet wird, ohne daß ein GOSUB-Statement durchlaufen wurde, da es sonst zu der Fehlermeldung RETURN WITHOUT GOSUB kommt. Dieser Fehler kann leicht vermieden werden, dadurch daß das Hauptprogramm mit *END abgeschlossen wird.

GOTO

Anweisung, die es in allen BASIC-Dialekten gibt. Durch ein GOTO-Statement wird ein *unbedingter Sprung ausgeführt, das heißt das Programm wird mit der Zeile fortgesetzt, die in dem GOTO Statement steht.

SYNTAX :

GOTO Zeilennummer

Der Sprung, der durch dieses Statement ausgeführt wird ist deshalb unbedingt, weil an die Sprunganweisung keine

Bedingung geknüpft ist und der Sprung somit auf jeden Fall ausgeführt wird.

Beispiel :

```
10 Z = Z + 1
20 PRINT Z, SQR(Z)
30 GOTO 10
```

In diesem Beispiel werden die Quadratwurzeln aller Zahlen errechnet, die der COMMODORE 64 überhaupt berechnen kann. Das GOTO-Statement in Zeile 30 ist an keine Bedingung geknüpft, das Programm springt also immer wieder zu Zeile 10 zurück.

Graphik

Der COMMODORE 64 kennt 3 verschiedene Arten der Graphik, 1. die Linien- und Blockgrapik, die über die Tastatur erreichbar ist, 2. die hochauflösende Graphik und 3. die *Sprites.
Die Linien- und Blockgraphik ist die einfachste Form der Graphik, die mit dem COMMODORE 64 erzeugt werden kann. Sie ist direkt über die Tastatur darstellbar. Dies geschieht, indem die Taste mit dem gewünschten Graphikzeichen gedrückt wird und gleichzeitig noch eine andere Taste und zwar die *COMMODORE Taste für das linke Graphikzeichen und die *SHIFT Taste für das rechte Graphikzeichen. Die Graphikzeichen können auch durch ihren *ASCII-Code auf den Bildschirm gebracht werden.
Die hochauflösende Graphik ist etwas schwieriger anzusteuern, da hier einige Register verändert werden müssen. Die SUPERGRAPHIK ist ein hervorragendes Hilfsmittel, um die Graphikfähigkeiten des COMMODORE 64 einfach und schnell zu benutzen.
Unter *Sprites versteht man bewegte Kleingraphiken, die zum Beispiel sehr häufig für Spiele benötigt werden. Der COMMODORE 64 ist in der Lage 8 Sprites gleichzeitig zu steuern.

Graphikzeichen

Elemente der Linien- bzw. Blockgraphik des COMMODORE 64. Auf fast jeder Taste der Tastatur befinden sich zwei dieser Graphikzeichen. Das linke Graphikzeichen kann durch geleichzeitige Betätigung der Taste und der *COMMODORE-Taste erzeugt werden, das rechte Graphikzeichen durch Betätigung der Taste und gleichzeitiger Betätigung der *SHIFT-Taste. Die *ASCII-Codes und die *Bildschirmcodes der einzelnen Graphikzeichen sind unter dem Buchstaben, auf dessen Taste sich die Grapikzeichen befinden aufgeführt.

Großrechner

Darunter versteht man Großcomputer. Die Großcomputer können zum einen durch ihre räumlich Ausdehnung und zum anderen durch ihre Leistungsfähigkeit von den *Personal-, den *Büro- und den *Homecomputern abgegrenzt werden. Großrechner sind zum Beispiel in der Lage mehrere verschiedene Programme, die in unterschiedlichen Programmiersprachen geschrieben sind, gleichzeitig zu bearbeiten. Außerdem zeichnen sie sich meist durch eine sehr hohe Speicherkapazität aus.

H

Taste der Tastatur des COMMODORE 64, mit der die Zeichen H, h, und dargestellt werden können. Der *Tastaturcode dieser Taste ist 29. Die *ASCII-Codes der Zeichen, die durch diese Taste dargestellt werden können, lauten wie folgt :

	Großschrift/Graphik	Groß-/Kleinschrift
H	72	200
h	-	72
	180	180
	121	-

Diese Zeichen haben außerdem die folgenden *Bildschirmcodes :

	Großschrift/Graphik	Groß-/Kleinschrift
H	8	72
h	-	8
	117	117
	72	-

Die Umschaltung zwischen den Modi Großschrift/Graphik und Groß-/Kleinschrift kann durch gleichzeitige Betätigung der Tasten *COMMODORE und *SHIFT vorgenommen werden.

Halbbyte

Darunter versteht man die Hälfte eines *Bytes, also genau 4 *Bits. Halbbytes ewrden benötigt, wenn *Daten gepackt dargestellt werden sollen.

Halbleiter

Dies sind chemische Elemente oder Verbindungen, deren elektrischer Wiederstand von der Temperatur abhängig ist und zwischen dem von Isolatoren und dem von Metallen liegt. Die Hableiter sind meist Kristalle, in seltenen Fällen auch Flüssigkeiten. Die Grundstoffe der Halbleiterentwicklung sind Silizium und Germanium. Für den Bau von Computern werden nur Halbleiter verwendet, deren Grundstoff Silizium ist. Halbleiter werden benötigt, um *Chips herzustellen.

Handbuch

Darunter wird die ausführliche Gebrauchsanweisung verstanden, die jedem Computer beiligt. Ein Handbuch sollte den gesamten Befehlsvorrat des Computers enthalten und die speziellen Syntaxregeln des Gerätes erläutern. Außerdem sollte es auf die spezifischen Eigenschaften des Computers, für den es erstellt wurde, eingehen. Handbücher gibt es auch für Programme, die so komplex sind, daß ihre Funktionen nicht mehr einfach erkennbar sind.

Handheld-Computer

Computer, die in einem Aktenkoffer untergebracht werden können und keine externe Stromversorgung brauchen, werden als Handheld-Computer bezeichnet. Solche Computer können zum Beispiel als mobile Einsatzstationen zur *Datenfernverarbeitung benutzt werden.

Handling

Handhabung. Darunter kann die Handhabung des Computers an sich verstanden werden, aber auch zum Beispiel die Handhabung der Bildschirmsteuerung, dann spricht man von Screen-Handling.

Hard-Copy

Wird eine Kopie des Bildschirminhaltes auf dem Drucker ausgegeben, so spricht man von einer Hard-Copy. Es können Hard-Copys von Texten angefertigt werden, bei dem COMMODORE 64 ist es mit Hilfe der SUPERGRAFIK auch möglich Hard-Copys von Bildschirmgrapiken anzufertigen.

Hard disk

Festplatte, *Magnetplatte, die nicht gewechselt werden kann, auf der Festplatte befinden sich alle wichtigen Teile des *Betriebssystems. Festplatten werden nicht für *Homecomputer angeboten, sonst sind sie für Computer aller Größenordnungen erhältlich.

Hardware

Alle Teile eines Computers, die in die Hand genommen werden können, werden als Hardware bezeichnet, also der Computer selbst und alle seine Peripheriegeräte wie zum Beispiel das Diskettenlaufwerk oder der Drucker. Das Gegenteil der Hardware ist die Software.

Hardwired

Festverdrahtet, darunter werden Funktionen verstanden, die nicht programmiert werden müssen, sondern stets benutzbar sind.

Hartsektoriert

Eigenschaft von *Disketten. Bei Disketten mit dieser Eigenschaft werden die Sektoren durch eingestanzte Löcher gekennzeichnet, diese Markierungen werden optisch, das heißt durch eine Lichtschranke, abgetastet. Bei hartsektorierten Disketten wird der vorhandene Speicherplatz besser ausgenutzt, als bei *softsktorierten Disketten.

Hauptspeicher

Synonymer Ausdruck für Arbeitspeicher. Dieser Speicher entspricht dem *RAM eines Computers. Er beinhaltet die Daten und Programme, die momentan verarbeitet werden. Wird das Gerät ausgeschaltet, so wird der Arbeitsspeicher gelöscht, er kann also nicht zur dauerhaften Speicherung von Daten und Programmen benutzt werden. Der COMMODORE 64 bietet für *BASIC grundsätzlich einen Arbeitsspeicher von ca. 38KB an und für die *Maschinensprache von ca. 52 KB.

Hexadezimalsystem

Zahlensystem dessen Basis die Zahl 16 ist. Das Hexadezimalsystem besteht aus folgenden Ziffern : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E und F.

dezimal	hexadezimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F
16	10

HEX$

Funktion des MIKROSOFT-BASIC. Diese Funktion wandelt eine dezimale Zahl in eine hexadezimale Zahl um.

SYNTAX :

HEX$(Zahl oder Variablenname)

Diese Funktion gibt es im BASIC des COMMODORE 64 nicht, sie kann aber über ein Programm ohne weiteres simuliert werden.

High level language

*Höhere Programmiersprache, dies sind Programmiersprachen, die nicht maschinenorientiert, sondern anwendungsorientiert sind.

HIRES

Abkürzung für "high resolution", gemeint ist damit die hochauflösende Graphik.

I

Taste der Tastatur des COMMODORE 64 mit dem *Tastaturcode 33. Durch diese Taste können die Zeichen I, ▬ i, ◹ dargestellt werden können. Diese Zeichen haben die folgenden *ASCII-Codes :

	Großschrift/Graphik	Groß-/Kleinschrift
I	73	201
i	-	73
▬	162	162
◹	105	-

Die *Bildschirmcodes dieser Zeichen lauten wie folgt:

	Großschrift/Graphik	Groß-/Kleinschrift
I	9	73
i	-	9
▬	98	98
◹	73	-

Die Umschaltung zwischen den Modi Großschrift/Graphik und Groß-/Kleinschrift kann durch die simultane Betätigung der Tasten *COMMODORE und *SHIFT vorgenommen werden.

IC

Abkürzung für "integrated circuit". Die deutsche Übersetzung dafür ist integrierter Schaltkreis. Die ICs sind die Grundbausteine eines Computers.

IEC-BUS

Darunter versteht man ein genormtes System von Bussen, das aus 8 Datenleitungen, 5 Steuerleitungen und 3 Quittungsleitungen besteht. Dieses Bus-System wird zur Verbindung des Computers mit seinen Peripheriegeräten benötigt. Über den IEC-Bus können bis zu 15 verschiedene Geräte an einen Computer angeschlossen werden.

IF...THEN

Befehl, der in allen BASIC-Versionen vorhanden ist. Das IF...THEN Statement dient der Kontrolle des Programmablaufs.

SYNTAX :

IF Bedingung THEN Ausdruck oder Zeilennummer

Ist die Bedingung, die in dem IF-Statement angegeben wird erfüllt, so wird die Anweisung, die hinter dem THEN steht, ausgeführt. Ist sie nicht erfüllt, so wird das nächste Statement abgearbeitet. Steht hinter dem THEN kein Ausdruck sondern eine Zeilennummer, so wird ein *bedingter Sprung ausgeführt.

Beispiel :

```
10 A = A + 1
20 PRINT A,A^2
30 IF A < 21 THEN 10
40 END
```

In diesem Beispiel sollen die Zahlen von 1 bis 20 samt ihrer Quadrate ausgedruckt werden. Ist der Zähler A kleiner als die Zahl 21, so ist die Bedingung der Zeile 30 erfüllt und es wird ein bedingter Sprung zur Zeile 10 ausgeführt. Erreicht A

den Wert 21 so ist die Bedingung der Zeile 30 nicht mehr erfüllt und es Wird das nächste Statement, also die Zeile 40, ausgeführt.
Wie oben gesagt, kann hinter dem THEN auch ein Ausdruck stehen, das sieht dann zum Beispiel so aus :

```
10 A = A + 1
20 PRINT A, A^2
30 IF A = 12 THEN PRINT "ERSTES DUTZEND VOLL"
40 GOTO 10
```

In diesem Beispiel wird, nachdem A den WERT 12 angenommen hat, die Meldung "ERSTES DUTZEND VOLL" auf den Bildschirm gebracht.
Der COMMODORE 64 läßt es nicht zu, daß IF...THEN Bedingungen verschachtelt werden, da er die ELSE Bedingung nicht kennt.

Immediate adressing

*Direkt Adressierung

Impact Printer

Drucker bei dem das Zeichen durch mechanischen Anschlag auf das Papier gebracht werden. Dieser mechanische Anschlag wird entweder durch Typen oder durch Nadeln ausgeführt. Ein wichtiges Merkmal des Impact-Printers ist, daß er ein Farbband besitzt. Alternativen zum Impact-Printer sind der *Thermodrucker und der *Tintenstrahldrucker.

Implementierung

Wird ein Programm auf einem Rechner zum erstenmal zum Laufen gebracht, so ist es implementiert worden. Implementierung kann auch mit "Verfügbar machen" übersetzt werden, ein Programm das auf einem Computer läuft, ist für den Besitzer bzw. den Benutzer dieses Computers verfügbar.

Increment

Erhöhung, wird ein Zähler in einem Programm hochgezählt, so spricht man von der Erhöhung des Zählers, das Gegenteil des Increment ist das *Decrement.

Indexregister

Teil des *Betriebssystems des COMMODORE 64, dieses Register wird benötigt um Tabellen abzuarbeiten. Dieses Register fungiert quasi als Zähler.

INITIALIZE

Befehl des *DOS der COMMODORE-Diskettenlaufwerke. Dieser Befehl dient der *Disketteninitialisierung.

SYNTAX :

PRINT# logische Filenummer,"INITIALIZE"

Dieser Befehl wird benötigt, wenn auf eine *Diskette mit *Direktzugriffsbefehlen zugegriffen werden soll und die Diskette die gleiche ID hat wie die vorhergehende Diskette.

Soll auf eine *Diskette direkt zugegriffen werden, so benötigt das *DOS die aktuelle *BAM der Diskette, um festzustellen welche Blocks belegt sind und welch Blocks frei sind.
Der Befehl INITIALIZE kann durch "I" abgekürzt werden, bevor er benutzt werden kann, muß der entsprechende File geöffnet worden sein. Die Befehlsfolge sieht dann zum Beispiel wie folgt aus :

```
OPEN 1,8,15,"I"
CLOSE 1
```

INKEY$

Befehl des MIKROSOFT-BASIC. Dieser Befehl liest ein Zeichen von der Tastatur ein.

SYNTAX :

Variablenname = INKEY$

Zu beachten ist hier, daß die Variable, auf die der Wert von INKEY$ zugewiesen werden soll, eine alphanumerische Variable sein muß.
Den Befehl INKEY$ gibt es im BASIC des COMMODORE 64 nicht, er kann aber sehr leicht simuliert werden und zwar durch den Befehl *GET. In diesem Fall muß keinen Zuweisung auf eine Variable mehr vorgenommen werden, es kann in dem GET-Statement direkt die Variable angegeben werden, auf die der eingelesen Wert zugewiesen werden soll, außerdem ist es auch möglich den Wert auf eine numerische Variable zuzuweisen, sofern sichergestellt ist, daß nur numerische Werte eingegeben werden. Das GET-Statement des COMMODORE 64 BASIC bietet also mehr Möglichkeiten, als das INKEY$ Statement.

Inoperativ program

Darunter versteht man ein Programm, das im Moment nicht ausgeführt wird, sondern ruht. Dies kann zum Beispiel durch eine Warteschleife erzeugt werden, oder beim *Multitasking der Fall sein.

INP

Befehl des MIKROSOFT BASIC, der die Eingabe von Werten über Eingabeports steuert.

SYNTAX :

INP(Portadresse)

Die Portadresse darf zwischen 0 und 255 liegen. Dieser Befehl ist nicht im BASIC des COMMODORE 64 enthalten, er kann auch nicht simuliert werden, da dies von der Hardwarebedingungen des 6510 her nicht möglich ist.

Input

1.Eingabe, Eingang. Alles was in einen Computer eingegeben wird, kann als Input bezeichnet werden, egal ob es über die Tastatur oder durch ein Peripheriegerät eingegeben wird.

2.Befehl, der in allen BASIC-Dialekten enthalten ist, seine Aufgabe ist es Daten vom Bildschirm einzulesen.

SYNTAX :

INPUT "Erläuterungstext";Variablenname-1,Variablenanme-2,...

Die eingelesenen Daten werden sofort den angegebenen Variablen zugeordnet. Der Erläuterungstext kann dazu benutzt werden Nachrichten an den Benutzer zu geben, zum Beispiel eine Erläuterung welche Daten eingegeben werden sollen.
Wird innerhalb eines Programms ein INPUT-Statement erreicht, so wird, falls vorhanden, der Erläuterungstext auf dem Bildschirm ausgegeben, anschließend wird ein Fragezeichen auf den Bildschirm gesetzt. Danach wartet der Computer auf die Eingabe, das Ende der Eingabe wird durch die Betätigung der *RETURN Taste gekennzeichnet. Innerhalb eines INPUT Statements, mit dem nur eine Variable erfasst werden soll, dürfen keine Kommata auftauchen, da der Computer sonst versucht die Zeichen, die rechts vom Komma stehen auf eine andere Variable zuzuweisen. Ist eine solch weitere Variable nicht im INPUT-Statement vorgesehen, so reagiert der COMMODORE 64 mit der Meldung "EXTRA IGNORED" und liest nur die Daten, die links vom Komma stehen. Diese Meldung führt aber nicht zu einer Programmunterbrechung.
Eine weitere Fehlerquelle bei der Arbeit mit dem INPUT Statement ist, daß als Variablentyp eine numerische Variable gewählt wird und während des Programmablaufs Buchstaben oder Sonderzeichen auf diese Variable zugewiesen werden. Der COMMODORE 64 bringt dann die Meldung "REDO FROM START" auf den Bildschirm und wiederholt das INPUT Statement. Ansonsten ist das INPUT Statement sehr einfach zu handhaben.

Beispiel :

```
10 PRINT CHR$(147)
20 INPUT "ZAHL     : ";Z
30 INPUT "EXPONENT : ";E
40 PRINT Z;" HOCH ";E;" = ";Z^E
50 GOTO 20
```

In diesem Beispiel wird das Ergebnis der Potenzierung einer beliebigen Zahl mit einem beliebigen Exponenten berechnet. Die Zahl und der Exponent werden durch INPUT Statements erfasst.

INPUT#

Befehl, der in fast allen BASIC-Versionen existiert, dieser Befehl liest Daten von Peripheriegeräten ein, also zum Beispiel vom *Diskettenlaufwerk oder von der *DATASETTE.

SYNTAX :

INPUT#logische Filenummer,Variablenname-1,Variablenname-2,...

Es gelten für diesen Befehl die gleichen Regeln, wie für den *INPUT Befehl, mit dem Unterschied, das hier kein Erläuterungstext verwendet werden kann, was auch überflüssig wäre, da das Einlesen der Daten nicht vom Benutzer beeinflusst werden kann und somit eine mnemotechnische Unterstützung nicht nötig ist.

INPUT$

Befehl des MIKROSOFT-BASIC, es läßt die Eingabe eines Zeichenstrings von genau festgelegter Länge zu.

SYNTAX :

INPUT$(Länge des Strings)

Es ist nicht möglich in diesem Statement die Variable direkt anzugeben, auf die der String zugewiesen werden soll. Dieser Befehl ist nicht im Befehlsvorrat des COMMMODORE 64 enthalten

und kann teilweise nur über einige Umwege simuliert werden. Sollen Strings, die nur ein Zeichen lang sind, eingelesen werden, so ist das eim COMMODORE 64 durch *GET möglich. Bei längeren Strings ist ein kleiner Trick erforderlich.

Beispiel :

```
10 INPUT A$
20 IF LEN(A$) > 10 THEN 10
30 ...
```

In diesem Beispiel wird in der Zeile 20 die Länge des Strings überprüft, ist der eingegebene String länger als vorgesehen, so muß die Eingabe wiederholt werden

Input device

Eingabegerät, darunter werden *Peripheriegeräte verstanden mit deren Hilfe Daten in einen Computer eingegeben werden können, solche Geräte sind zum Beispiel das *Diskettenlaufwerk, die *DATASETTE oder die *Tastatur.

Inquiry

Abfrage der Peripheriegeräte oder von Variablen.

INSTR

Befehl des MIKROSOFT-BASIC, dieser Befehl dient der Untersuchung von Strings. Mit seiner Hilfe kann ermittelt werden, ob ein Zeichen oder eine Zeichenfolge in einem String enthalten ist.

SYNTAX :

INSTR (Position, String, Suchstring)

Die Position in diesem Befehl gibt an von welcher Stelle an der String untersucht werden soll, "String" gibt den Namen des zu untersuchenden Strings an und "Suchstring" enthält das Zeichen bzw. die Zeichenfolge, nach der gesucht werden soll. Der Parameter "Position" kann auch weggelassen werden, dann wird der String von Anfang an untersucht. Das Ergebnis dieses Befehls ist eine Zahl und zwar Null, wenn der Suchstring nicht vorhanden ist und sonst die Stelle, an der der Suchstring beginnt. Der Befehl INSTR wird vom BASIC des COMMODORE 64 nicht angeboten, er kann aber mit gringem Aufwand simuliert werden.

Beispiel :

```
10 WOCHENTAGE$ = "MODIMIDOFRSASO"
20 INPUT "WOCHENTAG IN KURZFORM ";WT$
30 FOR I = 1 TO LEN(WOCHENTAGE$)
40 IF MID$(WOCHENTAGE$,I,2) = WT$ THEN PRINT "ALLES KLAR"
50 NEXT I
```

In diesem Beispiel wird die Eingabe daraufhin untersucht, ob sie in dem String "Wochentage$" enthalten ist. Auf diese Art und Weise kann der MIKROSOFT-BASIC Befehl simuliert werden, es muß nur in dem MID$ Befehl die Länge des Suchstrings angegeben werden.

Instruction

Befehl, Anweisung, darunter können allle *BASIC Schlüsselworte verstanden werden.

Instruction set

Befehlsvorrat eines Computers, der COMMODORE 64 hat einen Befehlsvorrat von ca. 80 BASIC-Befehlen und ca. 50 Maschinensprachebefehlen.

INT

Funktion die in allen BASIC-Versionen enthalten ist. Diese Funktion liefert den ganzzahligen Anteil einer Dezimalzahl.

SYNTAX :

INT (Zahl oder Ausdruck)

Diese Funktion kann benutzt werden, um unnötige Nachkommastellen einer Dezimalzahl abzuschneiden.

Beispiel :

```
10 INPUT "ERSTE ZAHL  ";A
20 INPUT "ZWEITE ZAHL ";B
30 C = A/B
40 C = (INT(C * 100))/100
50 PRINT "ERGEBNIS : ";C
```

In diesem Beispiel werden zwei Dezimalzahlen durcheinander geteilt, das Ergebnis wird mit zwei Nachkommastellen ausgegeben.

Integervariable

Numerische Variable, die nur als ganze Zahl verarbeitet wird. Integervariablen werden durch ein "%" hinter dem Variablennamen gekennzeichnet. Es ist zulässig Zahlen mit Nachkommastellen auf eine solche Variable zuzuweisen, es wird dann aber nur der ganzzahlige Anteil dieser Zahlen verarbeitet.

Interactiv mode

Bestimmte Form der Programmierung, darunter versteht man die *Dialogprogrammierung.

Interface

Synonymer Ausdruck für *Schnittstelle, darunter versteht man ein Zwischenstück, das zwei Geräte miteinander verbindet, die sonst nicht mit einander verbunden werden können.

Interpreter

Teil des Betriebssystems, der ein Programm, das in einer höheren Programmiersprache geschrieben wurde, direkt in die Maschinensprache übersetzt. Ein Interpreter geht das Programm Zeile für Zeile durch und übersetzt die jeweilige Zeile in die Maschinensprache, die übersetzte Zeile wird dann sofort ausgeführt. Der Vorteil eines Interpreters gegenüber einem *Compiler ist der höhere Komfort beim Programmtest, der Nachteil ist die erheblich langsamere Ausführungsgeschwindigkeit. Fast alle *Homecomputer arbeiten mit BASIC-Interpretern, da es erheblich einfacher ist einen Interpreter als einen Compiler zu erstellen. BASIC ist die

Proggrammiersprache, die am weitesten als Interpretersprache verbreitet ist, es gibt seltener auch *PASCAL-Interpreter.

Interrupt

Unterbrechung bzw. Unterbrechungssignal. Ein solcher Interrupt läuft beim COMMODORE 64 beispielsweise 60 mal innerhalb einer Sekunde ab, es wird dann das laufende BASIC-Programm unterbrochen und es wird die Tastatur abgefragt. Daher ist es möglich durch Betätigung der *RUN/STOP Taste das BASIC Programm abzubrechen.

Intervallschachtelung

Vorgehensweise der Mathematik, um Ergebnisse einer Funktion näherungsweise zu berechnen, die sich nicht direkt ermitteln lassen. Zum Beispiel kann die Ermittlung der Quadratwurtzel einer Zahl nur näherungsweise vorgenommen werden. Die Technik der Intervallschachtelung geht dabei so vor, daß zunächst relativ großzügig Unter- und Obergrenze des zu untersuchenden Intervalls festgelegt werden. Danach wird die Schrittweite gewählt, um die die Untergrenze vermehrt und die Obergrenze vermindert wird. Auf diese Art werden die Grenzen eines Intervalls immer enger gezogen bis eine vorher bestimmte *Genauigkeitsschranke erreicht worden ist.
Eine Intervallschachtelung besteht also aus einer endlichen Anzahl von einzelnen Intervallen, die ineinander geschachtelt werden.

Invalid

Ungültig, unzulässig

Iteration

Wird eine bestimmte Zahl von Programmstatements so oft durchlaufen, bis eine Bedingung erfüllt ist, so spricht man von einer Iteration. Durch eine *FOR-NEXT schleife ist eine sehr einfache Iteration zu erreichen. Andere Befehle, die eine Iteration erzeugen sind zum Beispiel : DO WHILE, REPEAT UNTIL, WHILE WEND usw.
Eine Iteration kann auch durch die Verknüpfung des Statements IF...THEN mit einem Sprungbefehl erzeugt werden.

Beispiel :

```
10 INPUT "NAME     : ";N$
20 INPUT "VORNAME : ";V$
30 IF V$ NOT = "ENDE" THEN 10
40 END
```

Hier wird solange ein Name eingegeben, bis für den Vornamen das Wort "ENDE"eingegeben wird. Die Iteration wird also so oft ausgeführt bis die obige Bedingung erfüllt ist.

J

Taste der Tastatur des COMMODORE 64 mit dem *Tastaturcode 34. Durch diese Taste können die Zeichen J, ▌ j, ▙ dargestellt werden. Diese Zeichen haben die folgenden *ASCII-Codes :

	Großschrift/Graphik	Groß-/Kleinschrift
J	74	202
j		74
▌	181	181
▙	106	-

Die *Bildschirmcodes dieser Zeichen lauten wie folgt:

	Großschrift/Graphik	Groß-/Kleinschrift
J	10	74
j	-	10
▌	117	117
▙	74	-

Die Umschaltung zwischen den Modi kann durch die gleichzeitige Betätigung der Tasten *COMMODORE und *SHIFT vorgenommen werden.

Job

Wird mit einem *Großrechner gearbeitet, so werden die einzelnen Programme "Job" genannt. Die Jobs stehen in einer Warteschlange, die nach bestimmten Kriterien abgearbeitet wird.

Joystick

Englischer Ausdruck für Steuerknüppel. Man versteht unter einem Joystick die Fernbedienung für einen Computer, diese Fernbedienung wird hauptsächlich für Spiele benötigt. Der COMMODORE 64 hat zwei Anschlüsse für Joysticks und zwar auf der rechten Seite der Konsole, die sogenannten *Control Ports.

Joystickabfrage

Der Joystick kann von einem Programm aus abgefragt werden. Dazu muß zunächst einmal die Tastatur gesperrt werden, dies geschieht durch POKE 56322,224. Der "Zustand" des Joysticks wird in der Speicheradresse 56320 für den Port2 und in der Speicheradresse 56321 für den Port1 abgelegt. Die vier Zustände haben folgende Werte :

Zustand	Wert
oben	1
unten	2
links	4
rechts	8

Folgendes Programm kann dazu benutzt werden die Joystickfunktionen zu testen.

```
10 POKE 56322,224
20 J = PEEK(56320)
30 IF (J AND 1) = 0 THEN PRINT "OBEN"
40 IF (J AND 2) = 0 THEN PRINT "UNTEN"
50 IF (J AND 4) = 0 THEN PRINT "LINKS"
60 IF (J AND 8) = 0 THEN PRINT "RECHTS"
70 IF (J AND 16) = 0 THEN PRINT "FEUERKNOPF"
80 GOTO 20
```

Dieses Programm testet den Joystick, der an den Port 2 angeschlossen ist, ist der Joystick an den Port 1 angeschlossen, so muß die Adresse in Zeile 20 auf 56321 geändert werden. Dieses Programm kann nur durch *RUN/STOP und *RESTORE beendet werden. Eingaben über die Tastatur können erst dann wieder vorgenommen werden. Das letzte Statement eines Programms, das mit den Joysticks arbeitet sollte POKE 56322,255 sein, da hiermit die Tastatur wieder freigegeben wird. Das Statement in Zeile 10 ist immer nötig, wenn eine Joystickabfrage erfolgen soll.

Jump

Englische Bezeichnung für *Sprung, Sprünge innerhalb eines BASIC Programms können durch verschiedene Befehle erzeugt werden, wie zum Beispiel *GOTO oder *GOSUB.

Justification

Darunter wird Blocksatz verstanden, dies ist ein Begriff der *Textverarbeitung, der Text wird in diesem Fall sowohl rechts- , als auch linksbündig ausgegeben.

K

Taste des COMMODORE 64 mit dem *Tastaturcode 37. Mit dieser Taste können die Zeichen K, k, ◧ und ◪ dargestellt werden. Diese Zeichen haben die *ASCII-Codes :

	Großschrift/Graphik	Groß-/Kleinschrift
K	75	203
k	-	75
◧	161	161
◪	107	-

Die *Bildschirmcodes dieser Zeichen lauten :

	Großschrift/Graphik	Groß-/Kleinschrift
K	11	75
k	-	11
◧	97	97
◪	75	75

Durch die simultane Betätigung der Tasten *COMMODORE und *SHIFT kann von dem Großschrift/Graphik Modus auf den Modus Groß-/Kleinschrift umgeschaltet werden.

Kanal

Hilfsmittel zur Übertragung von Informationen vom Computer zu den Peropheriegeräten und umgekehrt.

KB

Abkürzung für *Kilobyte

Key

Englische Bezeichnung für Taste.

Keyboard

Synonymer Ausdruck für *Tastatur

Keyword

Kennwort, eine andere Bezeichnung dafür ist *Password

KILL

Befehl des MIKROSOFT-BASIC. Dieser Befehl bietet die Möglichkeit aus einem Programm heraus Dateien von einer *Diskette zu löschen.

SYNTAX :

KILL Dateiname

Diesen Befehl gibt es im BASIC-Befehlsvorrat des COMMODORE 64 in dieser Form nicht, es ist aber auch mit dem COMMODORE 64 möglich aus einem Programm heraus Dateien von einer Diskette zu löschen und zwar auf folgende Weise :

PRINT#1,"S:dateiname"

Dies gilt aber nur falls vorher der Befehlskanal geöffnet worden ist, war dies noch nicht der Fall, so muß es nachgeholt werden.

Kilobyte

Maßeinheit der Speichergröße bzw. des Programmumfangs. Ein Kilobyte entspricht 1024 *Byte. Der COMMODORE 64 hat eine Hauptspeichergröße von 64 Kilobyte, wovon für BASIC-Programme 38 und für Maschinenspracheprogramme 52 Kilobyte zur Verfügung stehen.

Kommentar

Jedes gute Programm sollte Kommentare beinhalten, die einmal beschreiben was das Programm überhaupt leisten soll und zum anderen die Funktion einzelner Programmteile näher definieren. Weiterhin sollten durch Kommentare die Funktionen der Variablen erläutert werden. Das BASIC des COMMODORE 64 bietet einen Befehl an, der einen Text als Kommentar kennzeichnet, und zwar den Befehl *REM.

Kompatibilität

Von Kompatibilität wird dann gesprochen, wenn Dinge austauschbar sind. Softwarekompatibilität heißt dann, daß Programme zwischen zwei oder mehr Computersystemen ausgetauscht werden kann, Hardwarekompatibilität heißt das Geräte zwischen Systemen ausgetauscht werden kann.

Konfiguration

Darunter versteht man die Zusammenstellung der Komponenten eines Computersystems. Die Grundkonfiguration, mit der der COMMODORE 64 sinnvoll betrieben werden kann besteht aus dem COMMODORE 64, der DATASETTE und einem Fernsehgerät. Eine erweiterte Konfiguration könnte zum Beispiel aus dem COMMODORE 64, einem Diskettenlaufwerk, einem Drucker und einem Monitor bestehen.

Konjunktion

Operation der *Boolschen Algebra. Diese Operation verknüpft die Operanden durch die logische Operation "UND". Es können durch die Konjunktion zum Beispiel zwei oder mehr Bedingungen miteinander verknüpft werden.

Beispiel :

```
10 INPUT A
20 INPUT B
40 IF A > 30 AND B > 50 THEN GOTO 10
50 PRINT A * B
60 GOTO 10
```

In diesem Beispiel wurde die Bedingungen, daß A größer als 30 und B als 50 ist miteinander durch einen Konjunktion miteinander verbunden.

Konsole

Synonymer Ausdruck für *Tastatur

Konstante

Feststehender Wert innerhalb eines Programmes, der auch nicht verändert wird.

Beispiel :

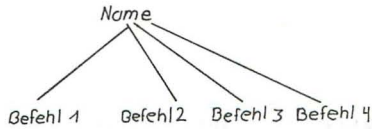
```
10 INPUT A
20 B = A * 2
30 PRINT B
40 GOTO 10
```

In diesem Beispiel wird der Wert 2 als Konstante benutzt und die Variable A wird mit der Konstanten multipliziert.

Kontrollbaum

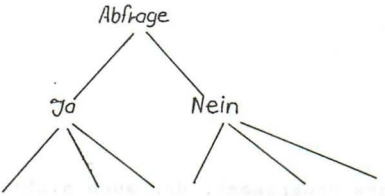
Hilfsmittel zum Programmentwurf, der Kontrollbaum dient dazu den Programmablauf übersichtlich darzustellen. Um Konntrollbäumme zu erstellen gibt es folgende Konstrukte :

1. die Sequenz



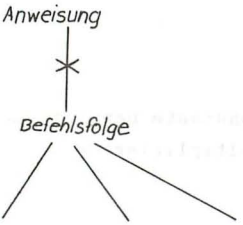
Die Sequenz stellt eine Folge von Befehlen dar, die nacheinander abgearbeitet werden.

2. die Selektion



Die Selektion stellt eine Programmverzweigung dar, wie sie zum Beispiel durch ein *IF...THEN Statement verursacht wird.

3.die Iteration



Unter einer *Iteration versteht man die Wiederholung einer Reihe von Statements und zwar so lange, bis eine bestimmte Bedingung erfüllt ist.
Eine Alternative zum Kontrollbaum als Hilfsmittel zum Programmentwurf ist das *Struktogramm und der Programmablaufplan.

Korrelationskoeffizient

Funktion der *deskriptiven Statistik. Ihre Aufgabe ist es ein Maß für den Zusammenhang zwischen den Ausprägungen zweier Variablen zu messen. Diese Funktion kann nur Werte zwischen -1 und +1 annehmen. Ist der Funktionswert -1, so sind die Variablen negativ korreliert, das heißt ihre Ausprägungen

verhalten sich gegenläufig, die Werte der Variablen A zum Beispiel steigen, währen die Werte der Variablen B fallen. Ist der Funktionswert 0, so besteht kein statistischer Zusammenhang zwischen den untersuchten Objekten. Nimmt die Funktion hingegen den Wert 1 an, so sind die Variablen positiv korreliert, ihre Werte verlaufen gleichsinnig.

L

Teil der Tastatur des COMMODORE 64 mit dem *Tastaturcode 42. Es können die folgenden Zeichen mit Hilfe dieser Taste dargestellt werden L, 1, ◨ und ◪ .
Die *ASCII-Codes dieser Zeichen lauten wie folgt :

	Großschrift/Graphik	Groß-/Kleinschrift
L	76	204
1	-	76
◨	182	182
◪	108	-

Den Zeichen können folgende *Bildschirmcodes zugeordnet werden :

	Großschrift/Graphik	Groß-/Kleinschrift
L	12	76
1	-	12
◨	118	118
◪	76	-

Die Umschaltung zwischen den Modi Großschrift/Graphik und Groß-/Kleinschrift kann durch gleichzeitige Betätigung der Tasten *COMMODORE und *SHIFT herbeigeführt werden.

Label

Marke, Kennzeichnung, gekennzeichnete Programmmstelle. Label werden vorallem bei der Maschinensprachprogrammierung benötigt, um Programmzeilen zu kennzeichnen, an die gesprungen werden soll. Auch in einigen *höheren Programmiersprachen wie zum Beispiel PL/1, COBOL oder PASCAL werden Programmzeilen zu diesem Zweck gekennzeichnet. Im

BASIC ist diese Möglichkeit grundsätzlich nicht vorgesehen, sie wird aber in einigen BASIC Erweiterungen, wie zum Beispiel SIMON's BASIC, angeboten.

Laden

Wird ein Programm oder eine Datendatei von einem externen *Massenspeicher in den Hauptspeicher des Computers transferiert, so sagt man zum Beispiel das Programm wurde geladen. Der Befehl, mit dem ein Programm geladen wird heißt *LOAD, das ist das englische Wort für laden.

Ladeprogramm

Besteht ein Programm aus zwei unabhängigen Programmen, von denen das erste die Aufgabe hat, das zweite von einem *Massenspeicher in den Hauptspeicher des Computers zu laden, so wird das erste Programm auch als Ladeprogramm bezeichnet.

Lader

Programm des *Betriebssystems, das Programme, die *absolut adressiert sind, von einem *Diskettenlaufwerk oder der *DATASETTE in den Hauptspeicher holt. Diese Programme werden an der korrekten Anfangsadresse abgelegt und können sofort gestartet werden.

Lag

Verzögerung, mit Time-lag wird zum Beispiel eine Zeitverzögerung bezeichnet.

Laufwerk

Verkürzte Bezeichnung für ein *Diskettenlaufwerk oder ein *Festplattenlaufwerk.

Leading zeroes

Führende Nullen.

Least significant bit

Darunter wird das *Bit mit dem niedrigsten Stellenwert bezeichnet.

Leerzeichen

Deutscher Ausdruck für Blank. Leerzeichen werden häufig zum Auffüllen von *Strings benötigt, oder zum Trennen von einzelnen Stellen. Leerzeichen können mit dem COMMDORE 64 durch die lange Taste am unteren Ende der Tastatur erzeugt werden. Diese Taste hat den *Tastaturcode 60. Dem Leerzeichen wird der *ASCII-Code 160 und die Bildschirmcodes 32 und 96 zugeordnet.

LEFT$

Befehl des MIKROSOFT-BASIC. Dieser Befehl gibt eine bestimmte Zahl von Zeichen aus, die von links aus einem *String gelesen werden.

SYNTAX :

LEFT$(Variablenname, Zahl)

Dieser Befehl ist auch im BASIC des COMMODORE 64 enthalten. Durch diesen Befhl ist es zum Beispiel möglich die fünf ersten Zeichen eines Strings zu lesen und zu verarbeiten.

Beispiel :

```
10 A$ = "COMMODORE 64"
20 PRINT LEFT$(A$,5)
```

Das Ergebnis dieses Programms ist "COMMO". Es ist auch möglich das Ergebnis diese Befehls auf eine andere Variable zuzuweisen und dann weiter zu verarbeiten.

LEN

Befehl des MIKROSOFT-BASIC, der auch im Befehlsvorrat des COMMODORE 64 vorhanden ist. Dieser Befehl dient der Ermittlung der Länge eines *Strings.

SYNTAX :

LEN(Variablenname)

Dieser Befehl ist sehr gut zu gebrauchen, wenn Strings Zeichen für Zeichen untersucht werden sollen und die Länge

der Strings nicht bekannt ist.

Beispiel :

```
10 INPUT A$
20 FOR I = 1 TO LEN(A$)
30 PRINT LEFT$(A$,I)
40 NEXT I
```

In nahezu allen Programmen, die Namen, Adressen und sonstige Texte verarbeiten wird dieser Befehl sehr häufig angewandt.

Lesen

Synomer Ausdruck für *Laden, dieser Ausdruck bezeichnet die Übertragung von Programmen und Daten von einem *Massenspeicher in den Computer.

Lernen, computergestütztes

Perfektionierte Form des programmierten Lernens. Das computergestützte Lernen ist eines der Hauptanwendungsbereiche der *Homecomputer, da die Homecomputer das Lernen sehr gut in kleine Spiele einbinden können, so daß das Lernen sehr abwechlungsreich gestaltet werden kann. Eine besondere Form des computergestützten Lernens ist das Lernen einer Programmiersprache, da hier das gelernte praktisch sofort am Objekt in die Praxis umgesetzt werden kann. Grundsätzlich wird unter computergestütztem Lernen das Lernen mit dem Computer verstanden, egal ob Vokabeln, eine Programmiersprache oder sonst etwas gelernt werden sollen.

LET

Befehl, den es in allen BASIC-Versionen gibt. Dieser Befehl nimmt die Zuweisung von Werten auf eine Variable vor.

SYNTAX :

LET Variablenname-1 = Variablenname-2 oder Ausdruck

Dieser Befehl ist zwar in allen BASIC-Versionen enthalten, er wird aber in den meisten Versionen nicht benutzt, da hier die Zuweisungen direkt vorggenommen werden können.

SYNTAX :

Variablenname-1 = Variablenname-2 oder Ausdruck

Der COMMODORE 64 nimmt die Zuweisungen direkt vor, das LET-Statement kann zwar benutzt werden, es ist aber nicht nötig.

Letter

Buchstabe

Library

***Bibliothek, darunter wird in Bezug auf die EDV eine Sammlung von Programmen verstanden**

Lichtstift

Peripheriegerät eines Computers. Der Lichtstift ist in der Lage Daten vom Bildschirm einzulesen, er tut dies in dem er einen Punkt des Bildschirms abliest und die Information über diesen Punkt an den Computer weitergibt, der dann mittels eines entsprechenden Programms darauf reagiert. Ein Lichtstifft ist zum Beispiel sinnvoll einsetzbar, wenn Konstruktionszeichnungen mit Hilfe eines Computers angefertigt werden sollen.

Lightpen

*Lichtstift

Lineare Optimierung

Mathematisches Hilfsmittel des Operations Research. Das Grundproblem der linearen Optimierung ist die simultane Lösung mehrerer linearer Gleichungen. Der Lösung mehrerer linearer Gleichungen liegt immer das gleiche Muster zu Grunde, so daß es sich anbietet die lineare Optimierung durch einen Computer vornehmmen zu lassen.

Linken

Darunter versteht man das Zusammenfügen verschiedener, unabhängiger Programme bzw. das Zusammenfügen eines Programms und der dazugehörigen Dateien. Bei der Arbeit mit dem COMMODORE 64 braucht nicht gelinkt zu werden. In der Regel wird das Linken erst bei *Bürocomputern nötig, da sie sehr häufig Mehrplatzanlagen sind, so daß genau gekennzeichnet

werden muß welche Programme zusammen gehören bzw. welche Dateien zu welchen Programmen gehören.

LINE INPUT

Befehl des MIKROSOFT-BASIC. Dieser Befehl dient der Erfassung von Zeichenketten.

SYNTAX :

LINE, INPUT "Eräuterungstext";Variablenname

Bei diesem Befehl ist darauf zu achten, daß als Variablenname nur der Name einer Stringvariablen benutzt werden darf. Der Vorteil dieses Befehls gegenüber dem *INPUT-Befehl ist, das die Zeichenketten nicht in Anführungsstriche gesetzt werden müssen, wenn sie Kommata oder führende Leerzeichen enthalten. Im BASIC des COMMODORE 64 ist dieser Befehl nicht enthalten, er kann in dieser Form auch nicht simuliert werden.

LINE INPUT#

Befehl des MIKRQSOFT-BASIC, der es ermögllicht eine ganze Textzeile aus einer Datei zu lesen. Eine Textzeile ist eine Folge von ASCII-Zeichen, die mit dem ASCII-Zeichen für *RETURN abgeschlossen worden ist.

SYNTAX :

LINE INPUT# Dateinummer, Variablenname

Dieser Befehl ist im BASIC des COMMODORE 64 nicht enthalten und kann auch nicht simuliert werden. Als Variablenname darf nur der Name einer Stringvariablen benutzt werden.

LIST

Systembefehl, den alle *Homecomputer kennen. Dieser Befehl dient dazu entweder das ganze Programm oder ausgewählte Programmteile auf dem Bildschirm oder dem Drucker auszugeben.

SYNTAX :

LIST	Gibt das ganze Programm aus.
LIST Zeilenr. -	Gibt das Programm von der angegebenen Zeile bis zum Ende aus.
LIST - Zeilennr.	Gibt das Programm bis zur angegebenen Zeilennummer aus.
LIST Zeilenr. - Zeilenr.	Gibt das Programm zwischen den angegebenen Programmzeilen aus.

Beim COMMODORE 64 wird die Ausgabe auf den Drucker durch den *CMD-Befehl eingeleitet.

Listing

Wird der Programmtext über den Bildschirm oder den Drucker ausgegeben, so erhält man ein Listing des Programms.

LLIST

Systembefehl, der das Programm oder ausgewählte Programmteile auf den Drucker ausgibt. Die Syntaxregeln sind hier die gleichen wie beim *LIST-Befehl. Die Ausgabe eines Programmlistings beim COMMODORE 64 wird durch den Befehl *CMD erzeugt.

LN

Mathematische Funktion, die den natürlichen Logarithmus einer Zahl ermittelt. Diese Funktion ist im BASIC des COMMODORE 64 enthalten und zwar unter der Bezeichnung *LOG.

LOAD

Systembefehl, der zur Übertragung von Daten und Programmen von einem *Massenspeicher in den *Hauptspeicher des Computers dient.

SYNTAX :

LOAD	Lädt das erste Programm einer Cassette.
LOAD "Programmname"	Lädt das Programm mit dem angegebenen Namen von der Cassette.
LOAD "*",8	Lädt das erste Programm der Diskette.
LOAD "Programmname",8	Lädt das Programm mit dem angegebenen Namen von der Diskette.

Dieser Befehl ist bei fast allen Computern in dieser Form verfügbar.

LOC

Abkürzung für "Lines of code". Darunter versteht man die Anzahl der Programmzeilen, die nicht der Dokumentation dienen, sondern ausführbar sind. LOC ist ein Maß für den Umfang eines Programms.

Location

Speicherzelle, bestimmte Stelle im *Hauptspeicher eines Computers, in die ein Wert gelegt werden kann.

Lochkarte

Medium zur Eingabe von Daten und Programmen in einen Computer. Da heute die Bildschirmterminals eine sehr große Verbreitung gefunden haben, werden fast keine Lochkarten mehr benutzt. Lochkarten bestehen aus dünner Pappe und sind in 10 Zeilen zu 80 Spalten eingeteilt. Gelocht wird eine Lochkarte mit einem Lochkartenstanzer, gelesen wird sie mittels eines Lochkartenlesers.

Lochstreifen

Veraltetes Medium zur Übertragung von Daten und Programmen in einen Computer. Lochstreifen bestehen aus Papier, in das nach

einem bestimmten Code Löcher gestanzt werden. Lochstreifen liegen auf Endlospapier vor. Gelesen wird ein Lochstreifen von einem Lochstreifenleser, gestanzt wird er von einem Lochstreifenstanzer.

Lösungsalgorithmus

Ein *Algorithmus, der zur Lösung des gestellten Problems führt. Das wichtigste Merkmal eines Programms ist der ihm zu Grunde liegende Lösungsalgorithmus.

LOG

Mathematische Funktion, deren Ergebnis der Logarithmus einer Zahl zur Basis 10 ist. Diese Funktion existiert im BASIC des COMMODORE 64 nicht. Durch folgende Reihe von Statements kann sie aber sehr leicht simuliert werden.

```
10 REM LZ = LOGARITHMUS ZUR BASIS 10
20 LZ = LOG(Zahl oder Ausdruck)/LOG(10)
```

Auf analoge Art und Weise können Logarithmen zu einer beliebigen Basis berechnet werden, es muß nur LOG(10) durch LOG(Zahl) ersetzt werden.

Logical add

Logische Addition, Verknüpfung zweier Variable durch die logische Operation UND.

Logical circuit

Logische Schaltung

Logic instruction

Logischer Befehl, dies kann zum Beispiel eine Vergleichsoperation mittels der *Boolschen Operatoren sein

Loop

Programmschleife

Low order

Niederwertig

LPOS

Befehl des MIKROSOFT-BASIC, der die aktuelle Position des Schreibkopfes beim Drucken angibt.

SYNTAX :

LPOS(beliebige Zahl)

Zu benutzen ist dieser Befehl bei der Steuerung des Druckers. Im BASIC des COMMODORE 64 ist dieser Befehl leider nicht enthalten und kann nur über große Umwege simuliert werden, es muß dann ständig mitgezählt werden, wieviel Zeichen und

wieviel Zeilen schon gedruckt worden sind.

LPRINT

Befehl des MIKROSOFT-BASC, der die Ausgabe auf den Drucker steuert. Für den LPRINT Befehl gelten die gleichen Regeln, wie für den *PRINT- Befehl.
Das BASIC des COMMODRE 64 beinhaltet den Befehl LPRINT nicht, es ist aber möglich ihn zu simulieren und zwar durch Kombination der Befehle *CMD und *PRINT.

LPRINT USING

Befehl des MIKROSOFT-BASIC, dieser Befehl dient der Steuerung der Druckerausgabe.

SYNTAX :

LPRINT "Formatangabe";Variablenliste

Die Formatangaben entsprechen denen der *PRINT USING Anweisung. Das BASIC des COMMODORE 64 enthält diesen Befehl nicht, er kann aber durch eine gut durchdachte Struktur der Druckerausgabe simuliert werden.

LSET

Befehl des MIKROSOFT-BASIC, dieser Befehl stellt Stringvariable linksbündig in den Datenpuffer ein.

SYNTAX :

LSET Name des Datenpuffers = Stringvariable

Im BASIC des COMMODORE 64 ist dieser Befehl nicht enthalten, er kann in dieser Form auch nicht simuliert werden.

M

Teil der Tastatur des COMMODORE 64 mit dem *Tastaturcode 36. Mit Hilfe dieser Taste können die folgenden Zeichen dargestellt werden : M, m, ▐ und ▚ . Diese Zeichen haben die folgenden *ASCII-Codes :

	Großschrift/Graphik	Groß-/Kleinschrift
M	77	205
m	-	77
▐	167	167
▚	109	-

Die *Bildschirmcodes dieser Zeichen lauten :

	Großschrift/Grapik	Groß-/Kleinschrift
M	13	77
m	-	13
▐	103	103
▚	77	-

Die Umschaltung vom Großschrift/Graphik Modus auf den Groß-/Kleinschrift Modus kann durch gleichzeitige Betätigung der Tasten *COMMODORE und *SHIFT vorgenmmen werden.

Machine code instruction

Maschinenbefehl

Machine language

*Maschinensprache

Machine program

Maschinenprogramm, darunter wird ein Programm verstanden, das in Maschinensprache geschrieben wurde.

Macro-instruction

*Makrobefehl

Magnetic head

Magnetkopf, Schreib- Lesekopf des *Diskettenlaufwerks oder des *Bandlaufwerks. Dieser Kopf gibt die Daten als magnetische Impulse an die Massenspeicher weiter.

Magnetic tape

*Magnetband

Magnetband

Externer Massenspeicher eines Computers. Von der Aufzeichnungstechnik her kann ein Magnetband mit einer Audiocassette verglichen werden, die Daten werden auf dem Magnetband sequentiell gespeichert, das heißt, es können auf einem Magnetband keine Dateien gespeichert werden, bei denen direkt auf spezielle Datensätze zugeriffen werden soll. Heute

werden Magnetbänder weitgehend nur noch zur Datenarchivierung verwendet. Ein Magnetband besteht, physikalisch betrachtet, aus einem Kunststoffband, dessen Oberfläche magnetisiert werden kann. In Bezug auf den COMMODORE 64 ist zu sagen, daß hier das Magnetband in Zusammenhang mit der *DATASETTE verwendet wird und zwar in Form einer normalen Audiocasette.

Magnetplatte

Die Magnetplatte ist quasi der große Bruder der *Diskette, sie besteht ebenfalls aus einer Kunststoffscheibe deren Oberfläche magnetisierbar ist. Die Magnetplatte hat die Funktion eines *Massenspeichers für einen Computer. Auf einer Magnetplatte sind in der Regel solche Informationen gespeichert, auf die sehr häufig zugegriffen wird, wie zum beispiel das *Betriebsystem oder Datensätze, die sehr oft benötigt werden. Bei den Magnetplatten unterscheidet man zwischen *Festplatte und Wechselplatte.

Mainframe

*Großrechner

Main storage

*Hauptspeicher oder Arbeitsspeicher eines Computers.

Makroassembler

Der Makroassembler ist in gewissem Sinne eine

Programmiersprache. Seine besondere Eigenschaft ist, daß hier ständig wiederkehrende Befehle unter einem Namen zusammengefaßt werden können. Im Programm wird dann statt der gesamten Befehlsfolge nur der Name dieser Sequenz aufgerufen.

Makrobefehl

Die Anweisung eines *Makroassemblers, die eine Reihe von Statements aufruf, ist ein Makrobefehl.

Manual

*Handbuch eines Computers oder eines Programmes.

Marke

Deutsche Übersetzung des Begriffs "Label", darunter wird eine symbolische Adresse verstanden; eine solche Adresse wird nicht direkt als absolute Zeilennummer oder direkt als Speicheradresse angegeben, sondern indirekt als Name. Im BASIC des COMMODORE 64 ist diese Möglichkeit grundsätzlich nicht vorgesehen, sie läßt sich nur mit sehr viel Aufwand simulieren, im SIMON'S BASIC hingegen ist diese Möglichkeit vorgesehen. Wird in einem BASIC Programm statt mit Zeilennummern als Sprungadressen mit Marken gearbeitet, so wird das Programm erheblich übersichtlicher und einfacher zu änderen.

Maschinensprache

Darunter versteht man eine *Programmiersprache, deren Befehle vom Computer direkt verstanden werden können, das heißt es wird weder ein *Interpreter noch ein *Compiler benötigt, um das Programm dem Computer verständlich zu machen. Daher sind Programme, die in Maschinensprache geschrieben worden sind auch erheblich schneller, als Programme, die zum Beispiel in BASIC geschrieben wurden. Ähnlich hohe Geschwindigkeiten wie bei Programmen in Maschinensprache lassen sich bei Programmen, die compiliert wurden, erreichen. Die Maschinensprache orientiert sich immer an dem Mikroprozessor, der mit ihrer Hilfe programmiert werden soll, und nicht am Benutzer. Die Maschinensprache des Prozessors 6510, der in den COMMODORE 64 eingebaut ist, unterscheidet sich deutlich von der Maschinensprache des Z80-Prozessors, der in einigen anderen *Homecomputern Verwendung findet und praktisch der Prozessor mit der weitesten Verbreitung ist. Ein hervorragendes Hilfsmittel zur Maschinenspracheprorammierung des COMMODORE 64 ist der PROFIMAT.

Maschinensprachemonitor

Die Aufgabe eines Maschinensprachemonitors ist es Speicher- und Registerinhalte anzuzeigen und die Möglichkeit zu bieten diese Inhalte zu ändern. Weiterhin ist das Abspeichern, das Laden und das Ausführen von Maschinenspracheprgrammen möglich. Ein komfortabeler Maschinensprachemonitor ist PROFI-MON 64.

Maske

Hilfsmittel der *Datenerfassung und der *Menuesteuerung von Programmen. Durch eine Maske wird der Bildschirm so

aufbereitet, daß der Benutzer ohne großen Aufwand mit dem Programm arbeiten kann, das heißt, es werden zum Beispiel Felder auf dem Bildschirm definiert, in die die zu erfassenden Daten eingetragen werden, diese Felder werden genau gekennzeichnet, so daß der Benutzer sehen kann, in welches Feld er welche Daten eintragen muß. Der Oberbegriff Maske kann in zwei Untergruppen eingeteilt werden und zwar in die Erfassungsmaken und die *Auswahlmasken. Erfassungsmasken wrden für die *Datenerfassung benötigt, während Auswahlmasken in der *Menuetechnik Verwendung finden.

Massenspeicher

Darunter versteht man Speicher mit einer sehr großen Kapazität. Massenspeicher sind zum Beispiel *Magnetband und *Magnetplatte. Die Massenspeicher des COMMODORE 64 sind die Cassette und die *Diskette. Die Kapazität eines Massenspeichers ist erheblich höher, als die Kapazität des *Arbeitsspeichers. Der Arbeitsspeicher des COMMODORE 64 hat für BASIC-Programme beispielsweise eine Kapazität von 38 KB, während eine Diskette mindestens eine Speicherkapazität von 170 KB hat. Außerdem bleiben die Daten auf einem Massenspeicher enthalten, während sie aus dem Arbeitsspeicher gelöscht werden, wenn die Stromversorgung unterbrochen wird. Massenspeicher werden hauptsächlich dazu benutzt, um Programme und Dateien zu archivieren.

Mass storage

*Massenspeicher

Master

1. Werden mehrere Computer zusammen als *Mehrplatzanlage betrieben, so ist immer einer der Computer der Master, das heißt von ihm aus können die anderen Computer überwacht und gesteuert werden. Der Master hat vor den anderen Computern beim Zugriff auf die gemeinsame *Peripherie immer Vorrang.

2. Hilfsmittel der Programmentwicklung auf den COMMODORE Computern. MASTER 64 ist quasi ein Programmierwerkzeug, es enthält die folgenden Elemente :

1. Bildschirmverwaltung zur raschen Erstellung komfortabeler *Bildschirmmasken
2. ISAM-Dateiverwaltung für schnellen Datenzugriff und effiziente Dateiverwaltung
3. Druck-Generator zum einfachen Erstellen und Austesten beliebiger Ausgabemasken
4. Programmschutz durch "NOLIST"-Modus und individuelle Schlüssel
5. Mehrfachgenaue Arithmetik, alle Grundrechenarten mit 22 Stellen Genauigkeit
6. BASIC-Erweiterungen; Toolkitfunktionen und das komplette COMMODORE-BASIC 4.0

Master processor

Hauptprozessor eines Computers, dieser Processor übernimmt die Steuerung der Systemkomponenten. Der Hauptprozessor des COMMODORE 64 ist der 6510.

Matrix

Synonymer Ausdruck für *Array. Matrizen werden hauptsächlich

in der linearen Algebra und in der Vektorgeometrie verwendet.

Matrix character

Matrixzeichen, dies ist ein Zeichen, das mit Hilfe einer *Matrix dargestellt wird.

Matrixdrucker

Drucker, der die Zeichen durch eine Matrix darstellt, meist werden 5*7 Matrizen verwendet. Häufig werden Nadeldrucker als Matrixdrucker verwendet, diese Drucker stellen die Zeichen als Punktmatrix dar. Die Zeichen werden also als Kombination einzelner Punkte dargestellt.

Matrix printer

*Matrixdrucker

MAX

Mathematische Funktion, die den größten mehrerer angegebener Werte ermittelt. Diese Funktion ist im BASIC des COMMODORE 64 nicht enthalten, sie kann aber durch ein kleines Programm leicht simuliert werden.

Beispiel :

```
10 DIM A(25)
20 FOR I = 1 TO 25
30 INPUT A(I)
40 NEXT I
50 MAX = A(1)
60 FOR I = 1 TO 25
70 IF A(I) > MAX THEN MAX = A(I)
80 NEXT I
90 PRINT "DIE GROESSTE EINGEGBENE ZAHL IST ";MAX
100 END
```

In diesem Beispiel werden 25 Zahlen willkürlich eingegeben und es wird die größte dieser Zahlen ermittelt. Es ist ohne weiteres möglich die Zahl, der zu untersuchenden Werte, willkürlich zu erhöhen, es muß nur darauf geachtet werden, daß der *Array richtig dimensioniert wird, und daß die Laufindizes richtig gesetzt werden.

MDT

Abkürzung für *mittlere Datentechnik

Mehrplatzsystem

Greifen zwei oder mehr Computer auf gemeinsame *Peripheriegeräte zu, so spricht man von einem Mehrplatzsystem. Einer der Computer diese Systems ist der *Master, dem die anderen Computer untergeordnet sind.

Mehrprogrammbetrieb

Werden auf einem Computer zwei oder mehr Programme gleichzeitig betrieben, so spricht man von Mehrprogrammbetrieb oder *Multitasking. Diese Technik wurde zuerst bei Großrechnern angewandt. Hier wird meist so vorgegngen, daß ein Programm eine gewisse Zeit lang bearbeitet wird, dann werden alle aktuellen Variablen gespeichert und das nächste Programm wird eine gewisse Zeit lang ausgeführt usw. Diese Vorgehensweise wird solange praktiziert, bis alle Programme abgearbeitet worden sind. Auch auf dem COMMODORE 64 ist der Mehrprogrammbetrieb in begrenztem Maße möglich, eine ausführliche Beschreibung dessen findet sich in dem Buch "64 Tips und Tricks".

Memory

Speicher

Memory address counter

Speicheradressenzähler

Memory bus

Speicherbus, dies ist ein Bus der die Daten zwischen Speicher und Arbeitsregister hin und herbewegt.

Memory data register

Speicherregister

MEMORY-EXECUTE

Befehl des *DOS der COMMODORE Diskettenlaufwerke. Dieser Befehl macht es möglich Maschinenprogramme im DOS aufzurufen und auszuführen, diese Maschinenprogramme können entweder Routinen des DOS-ROMs sein, oder eigene Maschinenroutinen sein. Die eigenen Routinen müssen mit "RTS" (Return from subroutine) abgeschlossen worden sein und durch den *MEMORY-WRITE Befehl zuvor in einen Pufferspeicher geschrieben worden sein. Der Befehl MEMORY-EXECUTE wird durch "M-E" abgekürzt.

SYNTAX :

"M-E" CHR$(LO) CHR$(HI)

Hier gibt "LO" das Lowbyte und "HI" das Highbyte an. Diese Angaben werden benötigt um die Startadresse des Maschinenprogramms eindeutig festzulegen.

MEMORY-MAP

Die Memory Map besteht aus einer Folge von Adresse und den dazugehörigen Funktionen. Sie kann als Speicherbelegungsplan aufgefasst werden, in dem alle Systemroutinen samt der dazugehörigen Einsprungadressen festgehalten sind. Mehr über die MEMORY-MAP findet sich in dem Buch "64 Intern".

Memory Protection

Speicherschutz, darunter wird zum Beispiel der Schutz vor unbeabsichtigtem Löschen verstanden. *Disketten lassen sich sehr leicht vor Überschreiben schützen und zwar durch den Schreibschutz.

MEMORY-READ

Befehl des *DOS der COMMODORE Diskettenlaufwerke, mittels dieses Befehls ist es möglich jedes Byte des DOS zu lesen. Abgekürzt wird dieser Befehl durch "M-R".

SYNTAX :

"M-R" CHR$(LO) CHR$(HI) CHR$(Anzahl)

"LO" und "HI" geben wieder das Low- und das Highbyte der Startadresse des zu lesendes Teils im DOS an, Anzahl gibt an, wieviel Bytes von der Startadresse an gelesen werden sollen.

MEMORY-WRITE

Befehl des *DOS der COMMODORE Diskettenlaufwerke, durch diesen Befehl ist es möglich direkt in das DOS-RAM zu schreiben. Abgekürzt wird dieser Befehl durch "M-W".

SYNTAX :

"M-W" CHR$(LO) CHR$(HI) CHR$(Anzahl) CHR$(Daten-1) CHR$(Daten-2)...

In "LO" und "HI" werden Low- und Highbyte der Startadresse angeben, in Anzahl die Zahl der Bytes die beschrieben werden

sollen und anschließend die Werte, die in die einzelnen Bytes geschrieben werden sollen. Durch diesen Befehl ist es zum Beispiel möglich die Gerätenummer des Diskettenlaufwerks zu änderen.

Menuesteuerung

Bestimmte Art und Weise des Programmaufbaus, der die Benutzersteuerung sehr erleichert. Die Grundlage der Menuesteuerung ist eine Abfolge von *Auswahlmasken, über die der Benutzer seine Anforderungen immer weiter spezifizieren kann. Wird die Menuetechnik angewandt, so sind zwei Grundforderungen zu erfüllen und zwar :

1. Es muß möglich sein aus jeder Maske heraus das Programm zu beenden
2. In jeder Maske muß die Möglichkeit gegeben sein in die vorhergehende Maske zurückzuspringen.

MERGE

1. Englischer Ausdruck für mischen

2. Befehl des MIKROSOFT-BASIC, der es zuläßt ein Programm, das sich im *Arbeitspeicher befindet, mit einem Programm, das sich auf einer Diskette befindet, zu mischen.

SYNTAX :

MERGE "Programmname"

Hierbei ist zu beachten, daß bei Programmzeilen, die beide

Programme aufweisen, die Zeilen des ersten Programms von denen des zweiten Programms überschrieben werden. Dieser Befehl existiert im BASIC des COMMODORE 64 nicht und kann auch nur durch sehr großen Aufwand simuliert werden. Es ist eher möglich durch Manipulation des Programmzeigers ein Programm zusätzlich in den Hauptspeicher zu laden.

MID$

Befehl des BASIC des COMMODORE 64. Dieser Befehl wird zu Verarbeitung von *Strings benötigt.

SYNTAX :

MID$ (Variablenname,Position,Länge)

Das Ergebnis diese Befehls ist ein Teilstring, der die in "Länge" angegebene Zahl von Zeichen enthält, beginnend an der Stelle "Position".

Beispiel :

```
10 INPUT A$
20 B$ = MID$(A$,3,5)
30 PRINT A$,B$
40 GOTO 10
```

In diesem Beispiel werden beliebige Strings eingegeben. Ausgegeben werden die Strings sowie Teilstrings, die von dem 3. Zeichen des Strings an 5 Zeichen enthalten. Die Werte für Position und Länge dürfen sich nur in einem Bereich zwischen 0 und 255 bewegen.

Seite 307

Mikrocomputer

Darunter versteht man Computer, die zwischen *Homecomputern und der *mittleren Datentechnik stehen, also die sogenannten Personalcomputer. Diese Computer können mit fast allen *Programmiersprachen betrieben werden, außerdem ist es möglich mit ihnen so aufwendig Programme wie zum Beispiel eine Finanzbuchhaltung oder ein Kalkulationsprogramm zu bearbeiten. Mikrocomputer haben in der Regel einen Hauptspeicherkapazität von mindestens 32 *KB. Die Programmiersprachen, die am häufigsten zur Programmierung von Mikrocomputern verwendet werden sind *BASIC und *PASCAL.

Mikroprozessor

Dies ist ein integrierter Schaltkreis, der quasi als Gehirn des Computers bezeichnet werden kann. Der Mikroprozessor ist die *CPU des Computers, er besteht aus den folgenden Bestandteilen :

1. Das Steuerwerk, es regelt die Übersetzung und die Ausführung der Befehle
2. Das Rechenwerk, seine Aufgabe ist die Verarbeitung der Daten
3. Das Speicherwerk, mit der Aufgabe den Transport und die Zwischenspeicherung der Daten zu überwachen.
4. Der Befehlssatz, er beinhaltet alle Befehle, die der Prozessor verarbeiten kann.

Das Kernstück des COMMODORE 64 ist ein Mikroprozessor vom Typ 6510.

MIN

Mathematische Funktion, die den Minimalwert einer Reihe von Zahlen angibt. Diese Funktion ist im BASIC des COMMODORE 64 nicht enthalten, sie kann aber durch ein kleines Programm sehr leicht simuliert werden.

Beispiel :

```
10 DIM A$(30)
20 FOR I = 1 TO 30
30 INPUT A$(I)
40 NEXT I
50 MIN = A$(1)
60 FOR I = 1 TO 30
70 IF A$(I) < MIN THEN MIN = A$
80 NEXT I
90 PRINT "DIE KLEINSTE EINGEGEBENE ZAHL IST ";MIN
100 END
```

In diesem Beispiel können 30 beliebige Zahlen eingegeben werden, aus diesen Zahlen wird dann die kleinste Zahl ermittelt. Es ist ohne weiteres möglich die Zahl der zu untersuchenden Werte zu verändern, es ist nur darauf zu achten, daß der *Array richtig dimensioniert wird, und daß die Laufindizes richtig ggesetzt werden.

Mittelwert

Maß der *deskriptiven Statistik, es gibt den Durchschnitt der zu berechnenden Werte an. Als Funktion ist die Berechnung des Mittelwertes nicht im BASIC des COMMODORE 64 enthalten, er kann durch ein kleines Programm aber leicht berechnet werden.

Beispiel :

```
10 PRINT "DURCHSCHNITTLICHE KOERPERGROESSE"
20 INPUT G
30 IF G = 999 THEN 70
40 A = A + 1
50 GES = GES + G
60 GOTO 20
70 MW = GES/A
80 PRINT "DIE DURCHSCNITTSGROESSE BETRAEGT : ";MW
90 END
```

Auf diese Art und Weise kann jeder beliebige Mittelwert berechnet werden.

Mittlere Datentechnik

Darunter versteht man Computersysteme, die zwischen den *Mikrocomputern und den *Großrechnern liegen. Das Hauptmerkmal der Anlagen der mittleren Datentechnik ist, daß sie in der Regel als *Mehrplatzsysteme ausgelegt sind. Anlagen der mittleren Datentechnik werden häufig auch als *Bürocomputer bezeichnet.

Mnemonisch

Technik der Gedächnisstütze. Die Gedächnisstütze geschieht zum Beispiel durch sprechende Variablenamen, bzw. durch sprechende Programmname.

Mode

*Betriebsart

Modem

Hilfsmittel zur *Datenferübertragung. Es ermöglicht die Übertragung digitalisierter Daten von einem mobilen Datenerfassungsgerät zu einem Computer bzw. von einem Computer zu einem anderen Computer mittels einer Telefonleitung.

Monitor

Perpheriegerät eines Computers, das als Ausggabegerät dient. An den COMMODORE 64 kann als Bildschirmeinheit sowohl ein normaler Fernseher, als auch ein Monitor angeschlossen werden. Der grundsätzliche Unterschied zwischen einem Fernseher und einem Monitor ist, daß der Monitor eine höhere Taktfrequenz hat als ein Fernseher und somit eine höhere Schärfe aufweist.

Most significant bit

Bit mit dem höchsten Wert

MS-DOS

Sehr weit verbreitetes *Betriebssystem, das nach dem *CP/M das häufigste ist. MS-DOS ist meist auf IBM Personalcomputern implementiert.

Multiplex

Darunter wird eine Art der Datenübertragung verstanden, bei der sich mehrere verschiedene Daten bzw. Datenquellen eine Leitung teilen. Die Aufteilung erfolgt beim Empfänger der Daten.

Multistatements

Werden mehrere Statements in einer Programmzeile codiert, so spricht man von einem Multistatement. Im BASIC des COMMODORE 64 werden die einzelnen Statements eines Multistatements durch ein ":" getrennt. Bei Multistatements ist zu berücksichtigen, daß die Statements nacheinander abgearbeitet werden, genau so als würden sie in verschiedenen Programmzeilen stehen.

Multitasking

*Mehrprogrammbetrieb

Multiuser system

*Mehrplatzsystem

N

Teil der Tastatur des COMMODORE 64 mit dem *Tastaturcode 39. Mit dieser Taste können die Zeichen N, n, ◨ und ◪ erzeugt werden. Diese Zeichen haben die folgenden *ASCII-Codes :

	Großschrift/Graphik	Groß-/Kleinschrift
N	78	206
n	-	78
◨	170	170
◪	110	-

Diese Zeichen haben die *Bildschirmcodes :

	Großschrift/Graphik	Groß-/Kleinschrift
N	13	78
n	-	13
◨	106	106
◪	78	-

Die Umschaltung zwischen den beiden Modi kann durch gleichzeitige Betätigung der Tasten *COMMODORE und *SHIFT erfolgen.

Nadeldrucker

Drucker, der statt Typen Nadeln verwendet, um die einzelnen Zeichen darzustellen.Nadeldrucker sind immer *Matrixdrucker. Bei den meisten Nadeldruckern ist es möglich jede Nadel einzeln anzusteuern, so daß durchaus ein eigener *Zeichensatz definiert werden kann.

Näherungsrechnung

Verfahren der Mathematik, um Ergebnisse zu berechnen, die nicht direkt ermittelt werden können. Näherungsrechnungen werden meist als *Intervallschachtelungen durchgeführt. Ein typischer Anwendungsfall von Näherungsrechnungen ist die Berechnung von Polynomen höheren Grades.

NAME

Befehl des MIKROSOOFT-BASIC, der es gestattet eine Datei umzubenennen.

SYNTAX :

NAME Alter Dateiname AS Neuer Dateiname

Die Möglichkeit des Umbenennens von Dateien ist mit dem COMMODORE 64 ebenfalls möglich, der Befehl hat nur eine etwas andere Syntax.

SYNTAX :

```
OPEN 1,8,15,"R:Neuer Dateiname=Alter Dateiname"
CLOSE 1
```

Hierbei ist darauf zu achten, daß nur Dateien umbenannten werden können, die ordnungsgemäß geschlossen wurden.

Nesting

Verschachtelung, das BASIC des COMMODORE 64 bietet die Möglichkeit *FOR...NEXT Schleifen zu verschachteln.

Network

*Netzwerk

Netzwerk

Darunter versteht man die Verbindung mehrerer Computer untereinander, die auch einen gegenseitigen Datenaustausch zuläßt, sowie den Zugriff auf gemeinsame *Peripheriegeräte erlaubt. Der COMMODORE 64 ist nicht netzwerkfähig.

NEW

Befehl, der in allen BASIC-Versionen existiert. Dieser Befehl löscht das Programm und alle benutzten Variablen aus dem *Arbeitsspeicher.

SYNTAX :

NEW

Dieser Befehl kann sowohl ausserhalb eines Programms verwendet werden, um den Hauptspeicher wieder für ein neues Programm frei zu machen, als auch innerhallb eines Programmes, um dafür zu sorgen, daß der Arbeitsspeicher nach Ablauf des Programms gräumt wird. Mit Hilfe dieses Befehls kann auch ein einfacher Programmschutz konstruiert werden.

Beispiel :

```
10 INPUT "KENNWORT";A$
20 IF A$ = "C-64" THEN 40
30 NEW
40 PRINT " SIE SIND ALS BENUTZER IDENTIFIZIERT "
50 ...
```

Nibble

Halbbyte, die vier Bits, die entweder ganz rechts oder ganz links in einem Byte stehen.

Nullen, führende

Bei sehr vielen Programmiersprachen bzw, einigen Computern werden die Zahlen mit führenden Nullen verarbeitet, das heißt eine Dezimalzahl, wie zum Beispiel 23, wird als 0023 beispielsweise verarbeit. Wieviele führende Nullen jeweils existieren, hängt von der Stellenzahl, die für die Variable deklariert wurde, ab. Der COMMODORE 64 arbeitet nicht mit führenden Nullen, so daß das zeitraubende Umsetzen der Ziffern entfällt.

Number system

Zahlensystem

Numerisch

Numerische Daten sind Ziffern und Zahlen, sowie Zahlen, die ein Vorzeichen oder den Dezimalpunkt enthalten. Numerische Zeichen sind die Ziffern von 0 bis 10.

O

Taste der Tastatur des COMMODORE 64 mit dem *Tastaturcode 38. Durch diese Taste können die Zeichen O, ▬ o, ▢ dargestellt werden. Diese Zeichen haben die folgenden *ASCII-Codes :

	Großschrift/Graphik	Groß-/Kleinschrift
O	79	207
o	-	79
▬	185	185
▢	111	-

Die *Bildschirmcodes lauten wie folgt :

	Großschrift/Graphik	Groß-/Kleinschrift
O	15	79
o	-	15
▬	121	121
▢	79	-

Die Umschaltung zwischen den Modi Großschrift/Graphik und Groß-/Kleinschrift kann durch simultane Betätigung der Tasten *COMMODORE und *SHIFT vorgenommen werden.

Object code

Wird ein Programm, das in einer *höheren Programmiersprache geschrieben wurde, durch einen *Compiler geschickt, so erzeugt dieser Compiler den Object Code des Programms. Der Compiler übersetzt das Programm in eine Form, die vom Computer direkt ausgeführt werden kann. Die übersetzte Form des Programms ist der Object Code des Programms. Die Form in der das Programm vom Programmierer eingegeben wird ist der

*Source Code des Programms. Das wesentliche Merkmal des Object Codes eines Programms ist, daß er direkt vom Computer ausgeführt werden kann.

OCT$

Befehl des MIKROSOFT-BASIC, mit der Aufgabe eine Dezimalzahl in einen *String umzuwandeln, der den oktalen Wert der dezimalen Zahl darstellt.

SYNTAX :

OCT$(Zahl oder Ausdruck)

Diesen Befehl gibt es im BASIC des COMMODORE 64 nicht, es ist aber möglich diesen Befehl durch ein Programm zu simulieren. Der Kern dieses Programms besteht in der Berechnung der oktalen Zahlen.

ON ERROR GOTO

Befehl des MIKROSOFT-BASIC, der zu einer ursachengemäßen Fehlerbehandlung dient.

SYNTAX :

ON ERROR GOTO Zeilennummer

Durch diesen Befehl ist es möglich auf jeden Fehler folgerichtig zu reagieren, da die Variable ERROR den Code des Fehlers enthält. Das BASIC des COMMODORE 64 enthält diesen Befehl nicht, durch ein Maschinenspracheprogramm ist dieser Befehl aber mit etwas Aufwand zu simulieren.

Offline

Sind in einem Computersystem Komponenten zeitweise nicht mit der Zentraleinheit verbunden, so sind diese Komponenten offline geschaltet.

Oktalsystem

Zahlensystem, dessen Basis die Zahl 8 ist. Dieses Zahlensystem enthält nur die Ziffern 0 bis 7.

Dezimalzahl	Oktalzahl
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	10
9	11
10	12
11	13
12	14
13	15
14	16
15	17
16	20

ON...GOSUB

Befehl, den es in fast allen BASIC-Dialekten gibt. Die Funktion dieses Befehls ist analog zu der des Befehls *ON...GOTO.

SYNTAX :

ON Variablenname GOSUB Zeilennummer-1, Zeilennummer-2,...

Der Befehl ON...GOSUB ist analog zu dem Befehl *ON...GOTO zu behandeln.

ON...GOTO

Befehl, der in vielen BASIC-Versionen existiert, er erlaubt gezielte Sprünge je nachdem, welchen Wert eine Variable angenommen hat.

SYNTAX :

ON Variablenname GOTO Zeilennummer-1, Zeilennummer-2,...

Hierbei ist zu beachten, daß die Variable nur ganzzahlige Werte, von eins an aufsteigend annehmen darf. Der Wert der Variablen darf auch nicht größer sein als die Zahl der Zeilennummern, die hinter dem GOTO angegeben worden sind. Nimmt sie den Wert Null oder einen Wert, der größer ist als die Zahl der Zeilennummern, an, so wird das Statement ausgeführt, das dem ON-Statement folgt.

Beispiel :

```
10 INPUT "ZAHL";Z
20 IF Z < 0 THEN A=1
30 IF Z > 60 THEN A=2
40 ON A GOTO 80,90
50 PRINT Z, SQR(Z)
60 GOTO 10
80 PRINT "ZAHL IST NEGATIV"
90 GOTO 10
100 PRINT "ZAHL IST ZU GROSS"
110 GOTO 10
```

In diesem Beispiel sollen nur die Quadratwurzeln der Zahlen 0 - 60 berechnet werden, je nach Art des Fehlers wird eine gesonderte Fehlermeldung ausgegeben.

Online

Zustand innerhalb eines Computersystems, wenn die Komponenten mit der Zentraleinheit verbunden sind. Das Gegenteil ist *Offline.

OPEN

Befehl, den es in allen BASIC-Dialekten gibt, er dient zum Öffnen von Dateien, das heißt er macht Dateien und Geräte erst betriebsfähig.

SYNTAX :

OPEN Filenummer, Gerätenummer, Sekundäradresse

Die Gerätenummern der einzelnen Geräte sind :

Tastatur	0
DATASETTE	1
Bildschirm	3
Drucker	4 oder 5
Diskettenlaufwerk	8 oder 9-15

Die Sekundäradresse braucht nur im Zusammenhang mit dem Drucker oder mit dem Diskettenlaufwerk angegeben zu werden. Im Zusammenhang mit dem Drucker gibt die Sekundäradresse beispielsweise an welche Schhriftart gewählt worden ist ind in Zuasmmenhang mit dem Diskettenlaufwerk welcher Puffer oder welcher Kanal benutzt werden soll.

Operand

Zahlen, Ziffern, Zeichen oder Ausdrücke mit denen eine Operation ausgeführt werden soll. Soll zum Beispiel eine Multiplikation zweier Zahlen durchgeführt werden, so sind die Zahlen die Operanden der *Operation Multiplikation. Soll *Strings miteinander verkettet werden, so sind diese Strings Operanden der Operation Verkettung.

Operating

Bei *Großrechener oder *MDT-Anlagen, an denen mehrere Benutzer arbeiten, ist es nötig den Zugriff der einzelnen Benutzer auf die *CPU und die gemeinsame *Peripherie koordiniert werden. Genauso müssen die Warteschlangen überwacht werden. Diese Tätigkeit des Überwachens und koordinierens wird Operating genannt.

Operating Manual

Handbuch, das sich auf die Bedienung des Computers bezieht, unabhängig von der Programmiersprache.

Operating System

*Betriebssystem

Operation, arithmetische

Unter arithmetischen Operationen versteht man die vier Grundrechenarten (+ - * /). Werden also zwei Zahlen miteinander multipliziert, so wird mit den Zahlen die Operation Multiplikation durchgeführt.

Operation, logische

Logische Operationen sind Operationen der *Boolschen Algebra, also eine der Operationen AND, OR, NOT.

Operator

1. Person die das *Operating durchführt.
2. Symbol, das zur Darstellung einer Operation dient.

OPTION BASE

Befehl, den es in sehr vielen BASIC-Versionen gibt. Dieser Befehl wird zur Verwaltung von *Arrays gebraucht.

SYNTAX :

OPTION BASE Zahl

Für Zahl können innerhalb dieses Befehls nur die Werte Null und Eins eingesetzt werden. Dieser Befehl legt fest welcher der kleinste Index eines Arrays ist. Wird als OPTION BASE 1 eingegben, so ist der keinste Index des Arrays 1, wird 0 angegeben, so ist der kleinste Index 0. Dieser Befehl existiert im BASIC des COMMODORE 64 nicht, hier wird immer mit 0 als kleinstem Index gearbeitet.

OUT

Befehl des MIKROSOFT-BASIC, dieser Befehl ist das genaue Gegenteil des *INP-Befehls. Der OUT-Befehl hat die Aufgabe einen Wert über einen Ausgabeport auszugeben.

SYNTAX :

OUT Portadresse, Zahl

Dieser Befehl ist im BASC des COMMODORE 64 nicht enthalten und kann auch nicht simuliert werden, da der COMMODORE 64 nicht die entsprechenden *Hardwarevoraussetzungenbesitzt.

Output

Darunter versteht man das Ergebnis eines Programms, in der

Regel sind das Listen, die von einem Drucker erstellt wurden. Das Ergebnis, das am Bildschirm angezeigt wird kann ebenfalls als Output bezeichnet werden. Wörtlich übersetzt heißt Output Ausgabe.

Overflow

Von einem Overflow wird dann gesprochen, wenn zum Beispiel innerhalb einer Rechnung eine Zahl ausgerechnet wird, die größer ist als die Zahl, die der Computer maximal verarbeiten kann. Weiterhin kann ein Overflow auftreten, wenn versucht wird etwas in den *Arbeitsspeicher zu laden und dafür zu wenig Platz vorhanden ist. Overflow bedeutet wörtlich übersetzt Überlauf.

Overlay

Wörtlich übersetzt heißt dies Überlagerung. Darunter ist eine bestimmte Programmiertechnik zu vertsehen, die dann angewandt wird, wenn mit Programmen gearbeitet wird, die zu groß sind, als daß sie komplett in den Hauptspeicher passen. In einem solchen Fall werden Programmteile auf eine Diskette ausgelagert und bei Bedarf nachgeladen.

P

Taste der Tastatur des COMMODORE 64 mit dem *Tastaturcode 41. Durch diese Taste können die Zeichen P, p, dargestellt werden. Diese Zeichen haben die *ASCII-Codes :

	Großschrift/Graphik	Groß-/Kleinschrift
P	80	208
p	-	80
	175	175
	112	-

Die *Bildschirmcoddes der Zeichen lauten :

	Großschrift/Gaphik	Groß-/Kleinschrift
P	16	80
p	-	16
	111	111
	80	-

Die Umschaltung zwischen den beiden Modi wird durch die simultane Betätigung der Tasten *COMMODRE und *SHIFT vorgenommen.

Pack

Verdichten von Daten

Packen

Daten werden gepackt, wenn zwei Ziffern in einem Byte codiert werden. Dies ist nur bei Ziffernn möglich, da bei der binären

Darstellung von Ziffern nur das zweite *Halbbyte unterschiedlich ist. Das BASIC des COMMODORE 64 bietet keinen besonderen Befehl zum Packen an, diese Technik ist aber trotzdem mit dem COMMODORE 664 realisierbar und zwar durch ein kleines Programm. Dieses Programm ist in dem Buch "64 für Profis" zu finden.

Paddle

Fernbedienung für den COMMODORE 64. Paddles treten immer paarweise auf, sie werden an einen der Joystickports angeschlossen, so daß insgesamt vier Paddles an den COMMODORE 64 angeschlossen werden könnnen.

Paralleldrucker

Darunter versteht man einen Drucker, der eine ganze Zeile auf einmal druckt. Gewöhnliche Nadeldrucker oder Typenraddrucker drucken nur ein Zeichen.

Parität

Anzahl der "l" in einer Bitgruppe, die übertragen wird. Von einer geraden Parität wird dann gesprochen, wenn die Anzahl der "l"innerhalb einer Bitggruppe gerade ist, analog gilt dies für die ungerade Parität.

Paritätsbit

Dieses Bit wird einer Gruppe von Bits angeschlossen, damit

diese Gruppe entweder eine gerade oder eine ungerade Anzahl von "1" enthält. Durch eine solche Festlegung ist es möglich die Datenübertragung zu kontrollieren. Wurde zum Beispiel eine gerade Parität festgelegt und ist nach Übertragung der Daten die Anzahl der "1" ungerade, so ist es zu einem Fehler bei der Datenübertragung gekommen.

Parity bit

*Paritätsbit

Partitioning

Aufteilung, sowohl von Speicherplatz, als auch von Programmen in Teilprogramme.

PASCAL

*Höhere Programmiersprache, die die *strukturierte Programmierung zuläßt. PASCAL hat seine besonderen Vorzüge in der systematischen Sprachstruktur, der leichten Erlernbarkeit der Grundelemente und der Möglichkeit der Programmstrukturierung. PASCAL wurde ursprüngllich als Sprache entwickelt, die zur Einführung in die Programmierung dienen sollte. Heute hat sie sich aus dieser Rolle aber weitgehend gelöst und wird in allen Bereichen des EDV-Einsatzes benutzt. Für den COMMODORE 64 ist ein PASCAL-Compiler enthältlich, und zwar unter dem Namen "PASCAL 64". Dieser Compiler erleichert den Einstieg in die Programmiersprache PASCAL und damit in die *strukturierte Programmierung.

Password

Darunter versteht man ein Kennwort. Kennworte lassen sich sehr gut benutzen, um Programme und Daten vor unbefugtem Zugriff zu schützen. Eine Möglichkeit dies zu tun ist zum Beispiel, am Anfang eines Programmes das Kennwort abzufragen und wenn es falsch eingegeben wird, das Programm automatisch zu löschen.

PEEK

Befehl der in viele BASIC-Dialekten enthalten ist. Mit Hilfe des PEEK-Befehls kann der Inhalt beliebiger Speicherstellen gelesen werden.

SYNTAX :

PEEK(Variablenname oder Zahl)

Das Ergebnis dieses Befehls ist immer eine Zahl zwischen 0 und 255. Der PEEK-Befehl hat sehr vielseitige Anwendungsmöglichkeiten, eine davon ist zum Beispiel die Tastaturabfrage.

Beispiel :

```
10 PRINT PPEK(203)
20 GOTO 20
```

Durch diese beiden Statements kann die Tastaturabgefragt werden. In Zeile wird der Wert der Taste, die gerade betätigt wird angezeigt. Wird keine Taste betätigt, so ist das Ergebnis des PEEK-Befehls der Wert 64.

PERFORMANCE TEST

Hilfsprogramm der COMMODORE Test-/Demodisketten. Es ermöglicht alle Funktionen des Diskettenlaufwerks zu testen. Dieser Test führt alle Zugriffsbefehle aus, nach jedem Zugriff wird der Fehlerkanal angezeigt, so daß genau festgestellt werden kann, welche der Funktionen des Diskettenlaufwerks nicht einwandfrei ausgeführt werden. Der PERFORMANCE TEST sollte nur mit Disketten ausgeführt werden, die entweder noch nicht benutzt worden sind oder nur unwichtige Daten enthalten. Dies ist wichtig, da die eingelegte Diskette *formatiert wird und somit alle vorhandenen Daten gelöscht werden.

Peripheral device

Gerät, das zur *Peripherie eines Computers gehört

Peripherie

Darunter versteht man alle Geräte, die an einen Computer angeschlossen werden können. Solche Geräte sind Drucker, Diskettenlaufwerk, DATASETTE, Joystick, Paddle, Lightpen, Bildschirm bzw. Monitor usw.

Personal Computer

Unter diesen Begriff fallen Computer in der Größenordnung vom *Homecomputer bis zum *Bürocomputer. Grundsätzlich kann man sagen, daß Personal Computer Einplatzsysteme sind, die mit relativ geringem Kostenaufwand angeschafft werden können und

direkt am Arbeitsplatz zu installieren sind.

PI

Element der Tastatur des COMMODORE 64, dieses Element befindet sich auf der Taste "^", diese Taste hat den *Tastaturcode 54. Der ASCII-Code des PI-Zeichens ist 126, der *Bildschirmcode ist 94 im Großschrift/Graphik Modus. Die Zahl PI wird benötigt um Kreise und Kreiselemente, sowie Kugeln und Kugelelemente zu berechnen.

PL/1

*Höhere Programmiersprache, die zu den *strukturierten Programmiersprachen zählt. Pl/1 liegt nur als Compilersprache vor, im Bereich der Personal Computer ist PL/1 kaum verbreitet. Entstanden ist PL/1 als Entwicklung der Firma IBM. Diese Programmiersprache vereinigt die Vorteile der Programmiersprachen *COBOL und *FORTRAN, sie enthält die gute Dateiverarbeitung von COBOL und die mathematischen Routinen von FORTRAN.

Plausibilitätstest

Ein solcher Test sollte in einem Programm immer dann durchgeführt werden, wenn die zu verarbeitenden Daten nur Werte innerhalb bestimmter Grenzen annehmen dürfen. Zum Beispiel sollte ein Plausibilitätstest dann durchgeführt werden, wenn in einem Programm mit Kosten gearbeitet wird, Kosten dürfen niemals negative Werte annehmen. Ein anderes Beispiel für einen Plausibilitätstest ist die Ermittlung des *COSINUS eines Winkels, hier ist zu beachten, daß Winkel nie

kleiner als Null sein können.

Plotten

Darunter versteht man das Zeichnen von Kurven, Zeichen, Ziffern und Buchstaben durch ein spezielle Peripheriegerät, den *Plotter. Der Unterschied zwischen einem Drucker und einem Plotter liegt darin, daß der Plotter langsamer ist, aber dafür die Fähigkeit hat, Linien durchgezogen darzustellen und auch die Möglickeit bietet Rundungen auch als solche darzustellen. Der Vorgang des Zeichnens wird als Plotten bezeichnet.

Plotter

Peripheriegerät eines Computers, der die Möglichkeit des Zeichnens bietet. Die Tätigkeit eines Plotters wird als *Plotten bezeichnet.

Pointer

Darunter versteht man Daten, die eine Adresse beinhalten, die benötigt wird, um andere Daten im Speicher zu lokalisieren.

POKE

Befehl, der in sehr vielen BASIC-Versionen vorkommt. Dieser Befehl manipuliert den Inhalt bestimmter Speicherstellen ganz gezielt, er erlaubt also gezielte Eingriffe in das *Adressregister.

SYNTAX :

POKE Adresse, Wert

Die Adresse kann in mehreren Formen angegeben werden, einmal als Zahl, oder als Ausdruck oder als Variable. Dies gilt auch für den Wert. Bei dem POKE-Befehl ist zu beachten, daß die Werte, die für Adresse und Wert angegeben werden können, direkt von dem Computer abhängig sind mit dem gearbeitet wird. Der Grund dafür ist, daß das Adressregister von Computer zu Computer verschieden aufgebaut ist.

Polnische Notation

Bestimmte Art arithmetische Operationen darzustellen. Üblicherweise werden die Operationszeichen, wie +, -, * und / , zwischen die *Operanden gesetzt. Bei der polnnischen Notation werden die Operationszeichen hinter die Operanden gesetzt.

Beispiel :

Normale Notation : (A+B)*(C+D)
Polnische Notation : AB+CD+*

POS

Befehl des BASIC des COMMODORE 64. Dieser Befehl gibt die Bildschirmzeile an, in der das nächste *PRINT Statement ausgeführt würde.

SYNTAX :

POS(Zahl)

Hierbei ist noch zu bemerken, das "Zahl" jeden beliebigen Wert annehmen darf.

PRINT

Befehl, den es in allen BASIC-Dialekten gibt. Die Aufgabe dieses Befehls ist Daten auszugeben und zwar auf den Bildschirm.

SYNTAX :

PRINT Variable, Zahl, String oder Ausdruck

Der PRINT-Befehl kann sehr vielfältig verwendet werden, allein durch die Trennungszeichen zwischen den Daten, die ausgegeben werden sollen, kann die Bildschirmausgabe gesteuert werden. Durch den ";" als Trennungszeichen werden die so getrennten Daten praktisch direkt hintereinander ausgegeben, währen durch das "," als Trennungszeichen der Bildschirm in Zonen eingeteilt wird, die Daten werden dann in die einzelnen Zonen gesetzt. Das PRINT-Statement ist weiterhin so variabel, daß alle Formen, in denen Daten vorliegen können über dieses Statement ausgegeben werden können.

Beispiel :

```
10 FOR I = 1 TO 25
20 PRINT "COMMODORE 64","COMMODORE 64";"COMMODORE 64"
30 NEXT I
```

PRINT#

BASIC-Befehl, der die Ausgabe von Daten auf die Peripheriegeräte steuert.

SYNTAX :

PRINT#Filenummer,Variablenliste

Wichtig ist, im Zusammenhang mit diesem Befehl, daß PRINT# bei dem COMMODORE 64 nicht durch ?# abgekürzt werden darf. Weiterhin ist zu beachten, daß der Kanal, der benutzt werden soll zuvor geöffnet worden ist.

Beispiel :

```
10 OPEN 1,4,4
20 FOR I = 1 TO 50
30 PRINT#1,I,SQR(I)
40 NEXT I
50 CLOSE1
```

In diesem Beispiel wird eine Liste der Zahlen von 1 bis 50 sowie deren Quadratwurzeln auf einem Drucker ausgegeben.Wenn ein Ausgabekanal nicht mehr benötigt wird, so sollte er geschlossen werden.

Printer

Drucker

PRINT USING

Befehl, den es in vielen BASIC-Dialekten gibt, er dient der formatierten Ausgabe von Daten über den Bildschirm.

SYNTAX :

PRINT USING Formatangabe, Variablenliste

Die Formatangabe ist, betrachtet man die Möglichkeiten, von Computer zu Computer verschieden, so daß hier nur ganz allgemeine Aussagen über die Formatangabe gemacht werden können. Durch die Formatangabe kann beispielsweise festgelegt werden, wieviele Stellen einer numerischen Variablen auf dem Bildschirm ausgegeben werden sollen. Durch die PRINT USING Anweisung ist es zum Beispiel ohne großen Aufwand möglich Zahlen rechtsbündig darzustellen. Das BASIC des COMMODORE 64 enthält die Anweisung PRINT USING nicht. Die Möglichkeiten dieses Befehls können aber durch die PRINT Anweisung in Zusammenhang mit den Anweisungen *SPC und/oder *TAB, sofern numerische Daten ausgegeben werden sollen durch Rundungen, simuliert werden. Zusätzlich wird in der Regel zur Simulation der PRINT USING Anweisung noch der Befehl *LEN benötigt.

PRINT# USING

Befehl des MIKROSOFT-BASIC, der die gleichen Eigenschaften hat, wie der *PRINT USING Befehl, dieser Befehl hat die Aufgabe die Daten auf einen Drucker oder ein Diskettenlaufwerk formatiert auszugeben.

SYNTAX :

PRINT# Filenummer USING Formatangabe, Variablenliste

Dieser Befehl existiert im BASIC des COMMODORE 64 nicht, er

kann aber ähnlich wie die *PRINT USING Anweisung simuliert werden.

Problemanalyse

Bevor mit der Erstellung eines Programms begonnen wird, sollte zuerst eine Analyse des dem Programm zugrundeliegenden Problems vorgenommen werden. Eine Problemanalyse beginnt zunächst mit der Sammlung aller Informationen, die für das Problem wichtig sind. Soll zum Beispiel ein mathematisch-wissenschaftliches orientiertes Programm erstellt werden, so müssen zunächst die benötigten Formeln zusammengestellt werden. Der nächste Schritt der Problemanalyse ist die Stukturierung des Problems, das heißt, es muß das Problem in Teilprobleme zerlegt werden und die Beziehung der Teilprobleme zueinander muß dargestellt werden. Der letzte Schritt der Problemanalyse ist die *Spezifikation, die quasi das Ergebnis der Problemanalyse darstellt.

Programm

Unter einem Programm versteht man eine wohlgeordnete Folge von Anweisungen. Handelt es sich um ein EDV-Programm, so ist die wohlgeordnete Folge von Anweisungen in eine Form, sprich Sprache, zu bringen, die ein Computer verstehen kann.
Von einem verschachtelten Programm wird dann gesprochen, wenn ein Programm aus mehreren *Unterprogrammen besteht, die sich unter Umständen auch gegenseitig aufrufen.

Programmablaufplan

Der Programmablaufplan ist die graphische Darstellung des

logischen Programmablaufes, er beinhaltet in Form von Symbolen die einzelnen Programmschritte und stellt sie übersichtlich dar. Ein Programmablaufplan sollte immer der erste Schritt der Programmerstellung sein, da er das effektive und fehlerfrei Programmieren unterstützt. Sollen später Änderungen oder Erweiterungen an dem Programm vorgenommen werden, so wird dies wesentlich erleichtert, wenn vorher ein Programmablaufplan erstellt wurde. Programmablaufpläne können in der Form eines *Kontrollbaums oder eines *Struktogramms erstellt werden.

Programmiersprache

Computer sind nicht in der Lage die menschliche Sprache zu verstehen und folgerichtig zu verarbeiten, daher wurden Mittel entwickelt, mit deren Hilfe dem Computer Anweisungen gegeben werden können, die er auch verstehen und verarbeiten kann. Die Programmiersprachen können in die maschinenorientierten und die höheren Programmiersprachen unterteilt werden. Die maschinenorientierten Programmiersprachen sind von den Möglichkeiten und Grenzen des jeweiligen Computers bzw. *Mikroprozessors abhängig. Die Maschinensprache ist von Computer zu Computer daher sehr verschieden. Die höheren Programmiersprachen sind weitgehend von dem Computer, auf dem mit ihnen gearbeitet werden soll, unabhängig. Höhere Programmiersprachen sind zum Beispiel *BASIC, *COBOL, *PASCAL, *FORTRAN, *ALGOL, *PL/1 usw. Diese Programmiersprachen können nicht direkt vom Computer ausgeführt werden, sondern sie müssen erst von einem *Interpreter oder einem *Compiler verarbeitet werden. Die höheren Programmiersprachen werden auch sehr häufig als problemorientierte Programmiersprachen bezeichnet. Dies ist der Fall, da die einzelnen Programmiersprachen oft auf bestimmte Anwendungsbereiche zugeschnitten sind, so ist *COBOL zum Beispiel zur Lösung kommerzieller Probleme konzipiert worden und *FORTRAN zur Lösung technisch-mathematischer Probleme.

Programmierung

Der Vorgang der Umsetzung eines Problems in eine *Programmiersprache wird Programmierung genannt.

Programmierung, strukturierte

Wird in einer strukturierten Programmiersprache, wie *PASCAL,*PL/1 oder *STRUKTO 64, programmiert, so spricht man von strukturierter Programierung. Der Vorteil der strukturierten Programmierung ist, daß aus dem Programmlisting die Zusammenhänge zwischen Programmverwzeigungen und Unterprogrammaufrufen deutlich wird. Ein weiterer Vorteil dieser Art der Programmierung ist, daß hier der Befehl *GOTO nicht benötigt wird, dieser Umstand trägt wesentlich dazu bei ein Programm übersichtlich zu gestalten und es wartungsfreundlich zu machen. Mit dem COMMODORE 64 ist es möglich strukturiert zu programmieren und zwar im Zusammenhang mit den Programmiersprache *PASCAL 64 und *STRUKTO 64.

Programm jump

Verzweigung innerhalb eines Programms

Programm step

Schritt innerhalb eines Programms

Programmoptimierung

Nachdem ein *Programm erstellt wurde, sollte versucht werden es noch effizienter zugestalten und zwar dadurch, daß die Laufzeit des Programms verkürzt wird, falls möglich, oder das versucht wird mit weniger Speicherplatz auszukommen als bisher.

Programmschleife

Wird eine Folge von Anweisungen mehr als einmal durchlaufen, so spricht man von einer Programmschleife. Es gibt im BASIC des COMMODORE 64 mehrere Möglichkeiten mit Programmschleifen zu arbeiten und zwar folgendermaßen :

1. durch *FOR...NEXT Schleifen
In diesem Fall wird eine Folge von Anweisungen so lange abgearbeitet, bis eine Variable einen bestimmten Wert erreicht hat.
2. durch Simulation des MIKROSOFT-Befehls *WHILE...WEND
Hier wird einen Folge von Anweisungen solange durchlaufen, bis ein bestimmter Zustand eingetreten ist.
3.durch einen *Endlosschleife
Dieser Fall tritt immer dann ein, wenn eine Folge von Anweisungen ständig, ohne *Abbruchbedingung bearbeitet wird.

Programmverzweigung, bedingte

Eine bedingte Programmverzweigung wird durch den *IF...THEN-Befehl erzeugt. Man spricht in diesem Fall von einer bedingten Programmverzweigung, da das Programm, in

Abhängigkeit von einer Bedingung an einer bestimmten Stelle fortgesetzt wird.

Prozessor

Einheit, die für die Steuerung des Computers verantwortlich ist. Jeder Computer beinhaltet einen Prozessor, der als Zentraleinheit fungiert. Der Prozessor des COMMODORE 64 ist der 6510. Ein weiterer bekannter Prozessor ist der Z80.

Prüfbit

Ein *Bit, das zusätzlich zu den informationsenthaltenden Bits übertragen wird, um festzustellen, ob es zu Übertragungsfehlern gekommen ist. Eine andere Bezeichnung für Prüfbit ist *Paritätsbit.

Puffer

Unter einem Puffer wird ein Speicher verstanden, der Daten vorübergehend aufnimmt, und zwar solange bis sie wieder gebraucht werden bzw. bis sie verarbeitet werden können. Puffer können sowohl durch die *Hardware als auch durch die *Software erzeugt werden. Hardwaremäßig erzeugte Puffer werden immer dann benötigt, wenn Geräte miteinander arbeiten, die unterschiedliche Verarbeitungsgeschwindigkeiten haben. Softwaremäßig erzeugte Puffer sind Variable, die Daten solange verwalten, bis sie benötigt werden.

Q

Taste der Tastatur des COMMODORE 64 mit dem *Tastaturcode 62. Durch diese Taste können die Zeichen Q, ◫ q, ● erzeugt werden, diesen Zeichen entsprechen die folgenden *ASCII-Codes :

	Großschrift/Graphik	Groß-/Kleinschrift
Q	81	209
q	-	81
◫	171	171
●	113	-

Die *Bildschirmcodes dieser Zeichen lauten :

	Großschrift/Graphik	Groß-/Kleinschrift
Q	17	81
q	-	17
◫	107	107
●	81	-

Die Umschaltung zwischen den Modi Großschrift/Graphik und Groß-/Kleinschrift kann durch gleichzeitige Betätigung der Taste *COMMODORE und *SHIFT vorgenommen werden.

Quellprogramm

Programm, das in einer höheren Programmiersprache geschrieben wurde und von einem *Compiler in Maschinencode übersetzt werden soll.

R

Taste der Tastatur des COMMODORE 64. Der *Tastaturcode dieser Taste ist 17. Außer dem Buchstaben R enthält sie folgende weitere Zeichen: r, ⊞ und ⊟ . Diese Zeichen haben folgende *ASCII-Codes:

	Großschrift/Graphik	Groß-/Kleinschrift
R	82	210
r	-	82
⊞	178	-
⊟	120	-

Diese Zeichen haben die folgenden *Bildschirmcodes:

	Großschrift/Graphik	Groß-/Kleinschrift
R	18	82
r	-	18
⊞	114	-
⊟	82	-

Die Umschaltung zwischen den beiden Modi kann durch gleichzeitige Betätigung der Tasten *SHIFT und *COMMODORE herbeigeführt werden.

RAM

Abkürzung für Random Access Memory (Speicher mit wahlfreiem Zugriff)
Mit diesem Begriff wird der Teil des Speichers bezeichnet, der sowohl gelesen als auch beschrieben werden kann. Dies ist der eigentliche Arbeitsspeicher des Computers. Hier werden sowohl Programm als auch Daten gespeichert. RAM-Speicher sind immer vom Strom abhängig. Dies bedeutet,

daß alle gespeicherten Inhalte verloren sind, sobald der Rechner ausgeschaltet wird.
Der COMMODORE 64 besitzt 64 Kilobyte RAM. Davon stehen für BASIC-Programme 38911 Bytes zur Verfügung, für Maschinenprogramme 52 Kilobyte.

READ

Befehl um Daten aus DATA-Statements zu lesen.

SYNTAX:

READ Variable,Variable,...

Dieser Befehl dient dazu, in DATA-Zeilen gespeicherte Informationen in Variablen zu überweisen. So können z.B. Konstanten, die nur in bestimmten Programmteilen benötigt werden, bei Bedarf eingelesen werden.

Beispiel:

```
10 RESTORE
20 READA$,B$,C$
30 PRINT A$;B$;C$
40 DATA "Heute","ist","Mittwoch"
50 GOTO 10
```

Der READ-Befehl erlaubt die Verwendung einer beliebigen Anzahl von Variablen. Diese können gemischt sein, d.h. sowohl String- als auch numerische Variablen dürfen gemischt verwendet werden.

Realvariable

Numerischer Variablentyp, der beim COMMODORE 64 einen Gültigkeitsbereich von -2.93873588E-39 bis 1.7014183E38 hat. Die interne Rechengenauigkeit beläuft sich auf 10 Stellen, von denen 9 angezeigt werden.

real time operation

*Echtzeitbetieb

Rechengenauigkeit

Die Rechengenauigkeit resultiert aus der Anzahl Stellen hinter dem Komma, mit denen der Computer rechnet. Es gilt: Je mehr Stellen, desto höher die Rechengenauigkeit.
Der COMMODORE 64 rechnet intern mit 10 Stellen hinter dem Komma. Dies ist zwar eine ganze Menge, aber trotzdem begrenzt. Dadurch lassen sich kleine Rechenungenauigkeiten leider nicht vermeiden. Dies ist ein Nachteil aller Computer, die naturgemäß mit dieser begrenzten Genauigkeit arbeiten, d.h. alle gängigen Homecomputer.
Ein gutes Beispiel für die Ungenauigkeit ist die laufende Addition von 0,1. Lassen Sie das folgende Programm einmal durchlaufen und sich überraschen:

Beispiel:

```
10 Z=-3
20 PRINTZ
30 Z=Z+0.1
40 GOTO20
```

Rechtsbündig

Dieser Begriff aus der Textverarbeitung bedeutet, daß ein Text so geschrieben wird, daß der rechte Rand bei allen Zeilen gleich ist. Dieses Druckverfahren wird auch Blocksatz genannt.

Grundsätzlich gibt es zwei Möglichkeiten, den rechten Randausgleich zu erreichen. Die einfache und am meisten verbreitete Art ist, die Stellen zwischen den Wörtern möglichst gleichmäßig mit Leerzeichen aufzufüllen. So erhalten die Wörter einen größeren Abstand zueinander.

Erheblich besser sieht die Proportionalschrift aus. Bei diesem Verfahren werden die Buchstaben und die Leerzeichen gleichmäßig so weit verbreitert, bis der rechte Rand stimmt. Dies ist aber ein sehr kompliziertes Verfahren und wird nur von sehr wenigen Druckern und Textverarbeitungssystemen unterstützt.

record

*Datensatz

Redundanz

Redundanz liegt immer dann vor, wenn Informationen oder Materialien öfter oder reichhaltiger vorhanden sind als notwendig.
Von redundanter Information spricht man in der Nachrichtentechnik, wenn unnötige Informationen gesendet werden. Diese unnötigen Informationen haben dann den Sinn,

die Datensicherheit zu erhöhen.
Redundante Speicherung liegt vor, wenn in einer Firma z.B. die Kundennummer sowohl in der Kundendatei als auch in der Bestelldatei als auch in der Rechnungsdatei vorhanden ist. Diesem Übel versucht man in jüngerer Zeit mit Hilfe von Datenbanksystemen beizukommen.
Sind in einer Computeranlage manche Teile mehrfach vorhanden, um bei Ausfall sofort Ersatz zu haben, so spricht man ebenfalls von einem redundanten System.

Referenzliste

Die Referenzliste zu einem Programm gibt genau an, was die einzelnen Programmteile wie machen. Außerdem wird eine genaue Aufstellung aller benutzten Variablen, Übergabeparameter und erlaubten Werte erstellt.
Von *Compilern erstellte Referenzlisten geben auch meist an, in welchen Zeilen welche Variablen verwendet werden.

Refresher

Damit bei *dynamischen Speicherbausteinen der Inhalt nicht verlorengeht, müssen diese in regelmäßigen Abständen wieder aufgefrischt werden. Diese Aufgabe übernimmt der sogenannte Refresher.

relocatable

frei im Speicher verschiebbar, relokatible Programme (meist Maschinenprogramme) sind vom Speicherplatz unabhängig, sie benutzen nur die *relative Adressierung. Dadurch laufen diese in jedem Speicherbereich.

REM

Befehl zur Kennzeichnung von Kommentaren

SYNTAX

REM Kommentar

Dieser Befehl erlaubt es, jede Art von Kommentaren in einer Zeile unterzubringen. Alles was sich hinter einem REM-Befehl in einer Zeile befindet, wird vom Programmablauf ignoriert. So ist es möglich, Erklärungen zu Programmschritten im Programm unterzubringen, ohne den Programmablauf zu stören.

Natürlich benötigen diese Kommentare Speicherplatz. So gilt es für jeden Programmierer abzuwägen zwischen der Notwendigkeit, das Programm zu dokumentieren und dem benötigten Speicherplatz.

RENUM

Befehl des Microsoft-BASIC zur Neunumerierung der Programmzeilen

SYNTAX:

RENUM Anfangszeilennummer,Endzeilennummer,Zeilenabstand

Dieser Befehl erlaubt es, einen beliebigen Bereich des Programms neu zu numerieren. Dies kann besonders dann sinnvoll sein, wenn Programmzeilen eingefügt werden müssen und kein Platz mehr vorhanden ist. Natürlich werden alle Sprungbefehle, die auf die umnumerierten Zeilen zeigen, mit verändert.

Im COMMODORE 64 ist eine solcher Befehl nicht vorhanden. Dieser ist auch nicht ganz einfach zu programmieren. Gute BASIC-Erweiterungen wie MASTER 64 enthalten aber diesen Befehl.

Reservekopie

Die auch oft Sicherheitskopie genannte Reservekopie soll es ermöglichen, auch weiterarbeiten zu können, wenn die Programmdiskette aus irgendeinem Grund einmal defekt sein sollte. Da von Programmen oft ganze Betriebe abhängen, sind solche Kopien für ein Unternehmen oft von lebenswichtiger Bedeutung.
Reservekopien von Datendisketten sollten in regelmäßigen Abständen angefertigt werden. Sollte eine Datendiskette einmal defekt sein, so müssen nur die Veränderungen seit der letzten Erstellung der Kopie neu eingegeben werden (*Backup).

Reset

Zurücksetzen des Computers
Durch einen Reset wird der Computer in den Einschaltzustand zurückversetzt. Manche Arten von Resets löschen dabei nicht den Programmspeicher, d.h. das Programm bleibt erhalten. Dies ist besonders praktisch, wenn ein Programm auf Grund eines Fehlers in einer Endlosschleife hängt. Dann ist die einzige Möglichkeit, die Kontrolle über den Rechner wiederzugewinnen, ein Reset. Wenn dabei das Programm nicht gelöscht wird, so ist dies natürlich besonders praktisch.

Beim COMMODORE 64 läßt sich ein solcher Reset durch das gleichzeitige Betätigen der STOP- und der RESTORE-Taste erzeugen. Damit können Sie nahezu jedes Programm

unterbrechen, wenn dieses diese Tasten nicht absichtlich gesperrt hat.

RESTORE

1.) BASIC-Befehl zur Zurücksetzung des Zeigers auf DATA-Zeilen

2.) Taste des COMMODORE 64 zur Ausführung eines *Reset

zu 1.) SYNTAX:

RESTORE

Dieser Befehl erlaubt es, den Zeiger auf die DATA-Zeilen auf den Start zurückzusetzen. Sollen DATA-Zeilen mit Hilfe des *READ-Befehls mehrfach gelesen werden, so muß vor jedem Durchlauf dieser Befehl ausgeführt werden.

zu 2.) Wenn die RESTORE-Taste zusammen mit der STOP-Taste gedrückt wird, so wird ein *RESET ausgeführt.

RETURN

Befehl zur Beendigung von Unterprogrammen

SYNTAX:

RETURN

Dieser Befehl veranlaßt den Rücksprung aus einem Unterprogramm. Der Programmablauf wird dann nach dem aufrufenden *GOSUB-Befehl fortgesetzt.

Trifft das Programm auf einen RETURN-Befehl, ohne daß vorher ein GOSUB-Befehl erteilt wurde, so erscheint die *Fehlermeldung "RETURN WITHOUT GOSUB ERROR".

Ein Unterprogramm sollte immer mit Hilfe des RETURN-Befehls verlassen werden. Wird ein Unterprogramm einfach mit einem GOTO-Befehl abgebrochen, so bleiben die Bytes, die als Information über die Rücksprungadresse auf dem *Stack gespeichert wurden, auf diesem stehen und blockieren dort Speicherplatz. Da der Stack nur eine sehr begrenzte Kapazität hat, ist er unter diesen Umständen sehr schnell voll: Die Folge ist die Fehlermeldung "OUT OF MEMORY ERROR".

Rundungsfehler

Durch Rundung verringert sich automatisch die Genauigkeit einer Zahl. Die Differenz zwischen der Original- und der gerundeten Zahl wird als Rundungsfehler bezeichnet.

RIGHT$

Befehl zur Erstellung eines Teilstrings

SYNTAX:

RIGHT$(Variable,Wert)

Dieser Befehl ermöglicht es, aus einem String ein beliebig langes Teilstück beginnend am Ende herauszukopieren oder von einem String ein beliebig langes linkes Stück abzutrennen.

Beispiel:

```
10 INPUT A$
```

```
20 B$=RIGHT$(A$,3)
30 PRINTB$
40 GOTO10
```

Bei diesem Programm werden immer nur die rechten 3 Buchstaben eines eingegebenen *Strings ausgegeben.

RND

Funktion zur Erzeugung einer *Zufallszahl

SYNTAX:

RND(1)

Diese Funktion erzeugt immer eine *Zufallszahl zwischen 0 und 1. Durch rechnerische Operationen lassen sich Zufallszahlen in beliebigen Bereichen erzeugen.

Beispiel:

```
10 FOR I=1 TO 100
20 A=INT(RND(1)*30)+20
30 PRINTA
40 NEXTI
```

Das obenstehende Programm erzeugt 100 Zufallszahlen in dem Bereich von 20 bis 50.

ROM

Abkürzung für Read Only Memory (Lesespeicher)

Wie der Name bereits andeutet, können ROM-Speicher nur gelesen werden. ROM-Bausteine lassen sich, sind sie einmal programmiert, nicht mehr ändern.

Im ROM-Speicher eines Computers ist immer mindestens eine Startprozedur gespeichert. Im COMMODORE 64 befinden sich sowohl das Betriebssystem und das BASIC in ROM-Bausteinen. Dies hat den enormen Vorteil, daß sofort nach dem Einschalten des Rechners in BASIC programmiert werden kann. Dadurch, daß sich beim COMMODORE 64 das ROM softwaremäßig abschalten läßt, bleibt trotzdem der Vorteil der Verwendbarkeit jeder Programmiersprache erhalten.

Routine

Als Routinen werden häufig benutzte *Utilities oder *Unterprogramme bezeichnet.

RS 232

Schnittstelle zur seriellen Datenübertragung.

Diese Schnittstelle läßt die verschiedensten Übertragungsformate zu. So können 7 oder 8 Bit-Datenwörter übertragen werden, außerdem lassen sich die Anzahl der Stoppbits und die Parität programmieren.

Diese Schnittstelle hat sich international als Standard durchgesetzt, so daß Daten zwischen fast allen Computern ausgetauscht werden können.

RSET

Befehl des Microsoft-BASIC zur rechtsbündigen Formatierung eines Strings.

SYNTAX:

RSET Feld-String=Variable

Die Variable wird durch diesen Befehl rechtsbündig in den Feldstring eingesetzt.

Beispiel:

```
10 A$="          "
20 B$="TEST"
30 RSET A$=B$
40 PRINT B$
```

Dies ergibt für B$:

```
      TEST
```

RUN

Befehl zum Starten eines BASIC-Programms

SYNTAX:

RUN

Rundungsautomatik

Die letzte Stelle, mit der der COMMODORE 64 intern rechnet, wird automatisch gerundet. Das BASIC des COMMODORE 64 besitzt keinen Befehl, der automatisch eine Zahl auf eine bestimmte Anzahl Stellen rundet. Dies läßt sich jedoch mit einem kleinen Programm leicht beheben:

Beispiel:

```
10 INPUTA
20 B=INT(A+0.5)
30 PRINTB
40 GOTO20
```

Die eigentliche Rundung findet in Zeile 20 statt. In diesem Beispiel wird auf eine ganze Zahl gerundet. Dazu wird zuerst einmal zu der Zahl 0.5 addiert. War die erste Nachkommastelle kleiner 0.5, so bleibt die Zahl vor dem Komma unverändert. War diese Stelle größer oder gleich 0.5, so erhöht sich die Zahl vor dem Komma um 1. Danach wird mit Hilfe des INT-Befehls der ganzzahlige Wert berechnet, d.h. die Stellen rechts vom Komma werden einfach vergessen. Die Zahl ist dann in der Variablen B gespeichert, wird ausgedruckt und dann die Eingabe der nächsten Zahl erwartet.

S

Taste der Tastatur des COMMODORE 64. Der *Tastaturcode dieser Taste ist 13. Außer dem Buchstaben S enthält sie folgende weitere Zeichen: s, ◱ und ♥ . Diese Zeichen haben folgende *ASCII-Codes:

	Großschrift/Graphik	Groß-/Kleinschrift
S	83	211
s	-	83
◱	174	-
♥	115	-

Diese Zeichen haben die folgenden *Bildschirmcodes:

	Großschrift/Graphik	Groß-/Kleinschrift
S	19	83
s	-	19
◱	110	-
♥	83	-

Die Umschaltung zwischen den beiden Modi kann durch gleichzeitige Betätigung der Tasten *SHIFT und *COMMODORE herbeigeführt werden.

SAVE

Befehl des BASIC des COMMODORE 64, der zum Abspeichern eines Programms dient.

SYNTAX:

SAVE Name, 0 oder 8

Dieser Befehl erlaubt die Abspeicherung BASIC-Programmen auf Kassette oder Diskette. Bei der Abspeicherung auf Kassette kann der Name weggelassen werden. Ebenfalls nicht notwendig ist das Komma und die 0.
Bei der Abspeicherung auf Diskette ist der Name, das Komma und die 8 obligatorisch. Dabei hängt die 8 allerdings von der Geräteadresse der Floppy ab. Haben Sie zwei 1541-Floppys angeschlossen, so hat die zweite zumeist die Geräteadresse 9. Dann muß der Befehl zur Abspeicherung eines Programms auf das zweite Diskettenlaufwerk lauten:

SAVE Name,9

Sollen Maschinenprogramme abgespeichert werden, so muß hinter die 8 noch ein ",1" gehängt werden.

Beispiel:

SAVE Name,8,1

Schnelldrucker

Druckertyp, der in der Lage ist, gleichzeitig eine gesamte Zeile zu drucken, z. B. Ketten-, Walzen- oder Trommeldrucker. (*Paralleldrucker)

Schnittstelle

deutsche Übersetzung des Wortes "Interface"

Verbindungsstelle eines Computers zur Außenwelt.

Dank Schnittstellen ist die Kommunikation mit Floppy, Drucker oder anderen Computern möglich. Über Schnittstellen werden Daten vom Computer zur Peripherie und umgekehrt übertagen.

Der COMMODORE 64 besitzt mehrere verschiedene Schnittstellen. Die serielle Schnittstelle dient als Verbindung zur Floppy 1541 und zu den Druckern.
In den Modulschacht läßt sich eine sogenannte IEC-Bus-Schnittstelle einstecken, die die Kommunikation mit den "großen" COMMODORE-Floppies ermöglicht.

An den Userport lassen sich selbstgebastelte Schnittstellen anschließen. So läßt sich diese als Eingang für Temperaturmessungen oder als Centronicsschnittstelle benutzen.

Schnittstelle; parallel

Bei dieser Art von Schnittstelle wird immer ein ganzes Wort gleichzeitig übertragen. Bei 8-Bit Rechnern wie dem COMMODORE 64 ist dies ein Byte = 8 Bit. Vorteil dieser Art der Übertragung ist die große Geschwindigkeit und die einfache Handhabung. Nachteil ist die Verwendung von vielen Leitungen (hier 9) und die damit verbundenen Kosten.

Schnittstelle; seriell

Bei dier Art von Schnittstelle werden die Daten Bit für Bit übertragen. Bei dieser Art der Übertragung werden nur zwei Leitungen benötigt. Der Vorteil der geringen Kosten wird mit Geschwindigkeitsverlust und der schwierigeren Handhabung erkauft.

Schreibtischtest

Der Schreibtischtest dient dazu, den korrekten Ablauf eines Programmes vor dem Eintippen in den Computer zu überprüfen.

Dabei nimmt man Werte, die bekannte Ergebnisse liefern und rechnet von Hand alle Formeln und Rechenwege durch. Dann werden die errechneten Ergebnisse mit den bekannten Werten verglichen. So lassen sich auf einfache Weise Formelfehler, aber auch logische Fehler im Programmablauf entdecken.
Beim Schreibtischtest überernimmt der Mensch soweit wie möglich die Funktion des Computers.

SCRATCH

Diskettenbefehl zum Löschen eines Files

SYNTAX

S:Name

Der SCRATCH-Befehl kann nur benutzt werden, wenn der Befehlskanal der Floppy geöffnet worden ist. Um ein Programm mit dem Namen "TEST" zu löschen, ist folgendermaßen vorzugehen:

Beispiel:

```
OPEN1,8,15,"S:TEST":CLOSE1
```

Es ist durch diesen Befehl möglich, gezielt Proramme oder Dateien zu löschen. Es können mehrere Files durch einen Befehl gelöscht werden.

Beispiel:

```
OPEN1,8,15,"S:TEST,TEST2":CLOSE1
```

Dieser Befehl löscht allerdings nicht das Programm selber auf der Diskette sondern nur den Eintrag im Inhaltsverzeichnis. So ist es mit einem geeigneten Programm möglich, das Löschen wieder rückgängig zu machen (siehe Das große Floppybuch)

screen

Bildschirm

Screen Management

Als Screen Management wird jede Behandlung des Bildschirms durch ein Programm bezeichnet.

Der COMMODORE 64 besitzt sehr effektive und einfach zu handhabende Möglichkeiten, den Bildschirm zu gestalten. So lassen sich alle Befehle, die den Bildschirm beeinflussen, anhand des PRINT-Befehls ausführen. Extra Befehle zur Positionierung des Cursors, zum Löschen des Bildschirms oder zur Auswahl der Farben sind nicht notwendig.

Im sogenannten Hochkommamodus werden alle Tastendrücke, die normalerweise Bestandteil des Bildschirmeditors sind und unmittelbare Effekte auf dem Schirm verursachen, nicht ausgeführt sondern als Befehl abgespeichert.

In diesen Modus gelangt man, indem nach einem PRINT-Befehl

ein Anführungszeichen (") gedrückt wird. Betätigt man nun die HOME-, DEL, die Cursor- oder eine der Farbtasten, so erscheint ein graphisches Symbol auf dem Schirm.

So ist es auf einfachste Weise möglich, den Cursor zu positionieren, den Bildschirm zu löschen und so weiter.

Der Hochkommamodus wird durch das Setzen eines weiteren Anführungszeichen oder durch Betätigen der RETURN-Taste wieder aufgehoben.

Unter Screen Management wird auch die Behandlungdes Bildschirms durch den Computer verstanden. Der COMMODORE 64 besitzt ein excellentes Screen Management, d.h. er baut den Bildschirm sehr schnell auf.
Dazu gehört natürlich auch der Bildschirmeditor, der es durch einfaches Überschreiben, Einfügen oder Löschen mitten auf dem Bildschirm erlaubt, Programmzeilen zu ändern.

SECANS

Trigonometrische Funktion, Umkehrfunktion zu COS(X)

Formel zur Berechnung des Secans:

SEC(X)=1/COS(X)

Diese Funktion ist im BASIC des COMMODORE 64 nicht enthalten, kann aber wie folgt simuliert werden:

Beispiel:

```
10 DEFFNSE=1/COS(X)
```

sector

Sektor, jede Diskette ist in *Spuren und Sektoren eingeteilt. Jede Spur ist dabei in eine bestimmte Zahl von Sektoren eingeteilt. Die Anzahl der Sektoren hängt vom Diskettenlaufwerk und der Position der Spur auf der Diskette an. Die äußeren Spuren enthalten mehr Sektoren als die inneren.

Seriell

Form der Datenübertragung.

Bei dieser Art wird jedes Bit nacheinander über eine Leitung geschickt. Vorteil dieser Art der Datenübertragung ist, daß nur zwei Kabel benötigt werden.
Nachteil gegenüber der parallelen Datenübertragung ist die relative niedrige Geschwindgkeit.

International hat sich als Standard die sogenannte *RS232- oder V24-Schnittstelle zur seriellen Datenübertragung durchgesetzt.

Auch für den COMMODORE 64 wird diese Schnittstelle angeboten. Sie wird an den Userport angeschlossen und softwaremäßig über Geräteadresse 2 angesprochen.

SGN

Mathematische Funktion zur Feststellung des Vorzeichens, sie ist expliziet im BASIC des COMMODORE 64 enthalten

SYNTAX:

Variable=SGN(Wert)

Diese Funktion liefert den Wert +1 bei einer positiven, den Wert -1 bei einer negativen Zahl und den Wert 0, wenn die Zahl 0 ist.

Beispiel:

```
10 INPUT A
20 B=SGN(A)
30 PRINT B
40 GOTO 10
```

Dieses kleine Programm liefert zu jeder Zahl den Wert 1, wenn sie positiv ist, zu jeder negativen Zahl den Wert 0, und bei Eingabe von 0 ebenfalls den Wert 0.

Silicon Valley

Silicon Valley ist der Name eines Tals südlich von San Francisco. Es darf getrost als der Geburtsort und das Zentrum der modernen Micro- und Personalcomputerindustrie bezeichnet werden.
Alle bedeutenden Hersteller haben in diesem Tal Ihren Firmensitz. Es gilt als das Mekka der Computerindustrie.

Simulation

Unter Simulation wird immer irgendeine Art von Nachbildung verstande. In der Computersprache versteht man darunter besonders die softwaremäßige Nachbildung einer bestimmten Hardware.

Oft werden komplexe Modelle in der Industrie, z.B.

Kraftwerke, in Computermodellen auf dem Bildschirm nachgebildet und alle ablaufenden Prozesse durchgerechnet. Simulationen werden auch in den Wirtschaftwissenschaften häufig benötigt, vor allem dann, wenn es sich um Rechnungen handelt, bei denen nicht alle Elemente bekannt sind. In diesem Fall wird das Eintreten bestimmter Ereignisse simuliert.

Im Zusammenhang mit der steigenden Verbreitung der EDV haben Simulationen an Bedeutung gewonnen, da sie in der Regel mit sehr viel Aufwand verbunden sind., der durch den Computer drastisch reduziert werden kann.

SIN

Trigonometrische Funktion zur Berechnung des Sinus einer Zahl, diese Funktion ist expliziet im BASIC des COMMODORE 64 enthalten.

SYNTAX:

Variable=SIN(Wert oder Ausdruck)

Das Ergebnis dieser Funktion wird in der Einheit Radiant (im Bogenmaß) ausgegeben.

Beispiel:

```
10 INPUT A
20 B=SIN(A)
30 PRINTB
40 GOTO 10
```

single board computer

Einplatinencomputer, Rechner, bei dem alle Bauelemente sich auf einer Platine befinden. Der COMMODORE 64 ist ein Einplatinencomputer.

Single Density

einfache Dichte

Aufzeichnungsformat für Disketten. Preiswerte Laufwerke speichern in diesem Format. Die Kapazität liegt bei ca. 200 KB.

single sided diskette

einseitige Diskette, diese kann nur auf einer Seite beschrieben und gelesen werden.

single user system

Einplatzsystem, Rechner, an dem nur eine Person arbeiten kann.

Softsektoriert

In jeder Diskette befindet sich das sogenannte Indexloch. Dies kennzeichnet den Beginn der Spur 0.

Die weitere Numerierung der Spuren und Sektoren wird

softwaremäßig bei der Formatierung der Diskette vorgenommen. Vorteil dieses Verfahrens ist, daß die Hersteller der Disketten sich nicht um das Aufzeichnungsformat kümmern müssen. Dadurch können sie für alle Firmen die gleichen Disketten herstellen und so den Preis niedrig halten.

Gegenteil : *Hardsektoriert

Software

Bezeichnung für alle Programme eines Rechners. Dazu gehören Betriebssystem, alle Dienst- und Anwenderprogramme.
Im Gegensatz dazu steht der Begriff Hardware für alle Bauteile des Rechners.

Software Entwicklungssystem

Computer mit Speichermedium und Drucker, der zum Entwickeln von Programmen dient.

Sonderzeichen

So werden alle Zeichen bezeichnet, die nicht zu den Zahlen und zu den Buchstaben des Alphabets gehören.
Und diese Sonderzeichen stehen Ihnen auf dem COMMODORE 64 zur Verfügung:

Zeichen	ASCII-Code	Bildschirmcode
!	33	33
"	34	34
§	35	35

$	36	36
%	37	37
&	38	38
'	39	39
(	40	40
)	41	41
*	42	42
+	43	43
,	44	44
-	45	45
.	46	46
/	47	47
:	58	58
;	59	59
<	60	60
=	61	61
>	62	62
?	63	63
	64	64
[	91	27
£	92	28
]	93	29
↑	94	30
←	95	31
	99	64
	123	91
	124	92
	125	93
	127	95
	164	100
	166	102
	168	104
	169	105
	186	122

Grafikzeichen sind in dieser Tabelle nur aufgeführt, wenn sie noch nicht bei der Erklärung der einzelnen Buchstaben

aufgetaucht sind.

Sortieren

Sortieren von Daten gehört zu den Aufgaben, die Computer besonders häufig, aber auch besonders effektiv ausführen. Es existieren die verschiedensten *Algorithmen, die alle ihre Vor- und Nachteile haben. Manche arbeiten bei völlig unsortierten Daten recht effektiv, bei Daten, die aber bereits teilweise in geordneter Form vorliegen, recht langsam.

Sortierroutine

Im folgenden ein kleines Sortierprogramm. Es ist durchschnittlich schnell, sehr einfach aufgebaut und zu verstehen.

Beispiel:

Das Programm sortiert ein beliebiges Stringarray. Zuerst das Listing:

```
100 AN=100
110 Z=0
120 IF INT(AN/2)<> AN/2 THEN Z=1
130 DIM A$(AN),B$(AN)
140 FOR I=1 TO AN
150 FOR J=1 TO 6
160 A$(I) = A$(I) + CHR$(RND(1)*26+65)
170 NEXTJ
180 NEXTI
190 FOR I=1 TO AN : PRINT A$(I), : NEXTI
200 PRINT "ES GEHT LOS"
210 TI$="000000"
```

```
220 FOR I=1 TO AN/2 + Z
230 GR$="          "
240 KL$="ZZZZZZZZZZZ"
250 Z=0
260 FOR J=1 TO AN
270 IF A$(J)="-1" THEN 300
280 IF A$(J)<KL$ THEN KL$=A$(J):ZK=J:Z=1
290 IF A$(J)>GR$ THEN GR$=A$(J):ZR=J:Z=1
300 NEXTJ
310 IF Z=0 THEN I=AN : GOTO 360
320 B$(I) = KL$
330 B$(AN+1-I) = GR$
340 A$(ZK)="-1"
350 A$(ZR)="-1"
360 NEXTI
370 PRINT TI$,TI
380 FOR I=1 TO AN
390 PRINT B$(I),
400 NEXTI
```

Die Zeilen 100 bis 190 dienen nur der Vorbereitung. Ein *Array wird dimensioniert und zufällig werden 100 Strings erzeugt. Die Anzahl der Strings wird in Zeile 100 festgelegt.
In Zeile 210 wird die "Stoppuhr" auf 0 gesetzt.

Die Sortierroutine sucht aus dem Array A$ den kleinsten und den größten Wert raus. Diese werden dann in das Feld B$ abhängig von dem Zähler I überwiesen. Um Zeit zu sparen, werden die rausgesuchten Felder in Zeile 340 und 350 ungültig gemacht.

Sound

Der Commodore 64 besitzt ein Sound-IC mit drei Tonkanälen

über 8 Oktaven. Jeder dieser Kanäle ist in der Hüllkurve frei programmierbar. Außerdem sind noch diverse Filter vorhanden (*Synthesizer)

Source Code

Quellprogramm

Bezeichnung für die Programmversion, die von einem *Compiler meist in ein Maschinenprogramm übersetzt wird.

space

Zwischenraum, auch Leerzeichen, die Leertaste wird auch als Space-Taste bezeichnet.

SPACE$

Befehl des Microsoft-BASIC zur Erzeugung einer Zeichenkette, die Leerzeichen enthält.

SYNTAX:

Variable=SPACE$(Wert)

Es wird ein String der Länge Wert erzeugt, der nur Leerzeichen enthält.
Im BASIC des COMMODORE 64 läßt sich dieser Befehl recht einfach simulieren:

Beispiel:

```
10 Z$=""
20 FOR I=1 TO 3
30 Z$=Z$+" "
40 NEXTI
```

SPC

BASIC-Befehl um eine bestimmte Anzahl Leerzeichen zu überspringen

SYNTAX:

PRINTSPC(Wert)

Dieser Befehl läßt sich nur in Zusammenhang mit dem PRINT-Befehl ausführen. Er bewirkt sowohl auf dem Bildschirm als auch auf dem Drucker,daß der Cursor sich um soviel Zeichen nach rechts bewegt wie die Zahl "Wert" angibt.

Beispiel:

```
10 PRINTSPC(3);"TEST";SPC(5);"TEST"
```

Dies ergibt:

```
   TEST     TEST
```

Dieser Befehl dient der Formatierung der Ausgabe sowohl auf dem Bildschirm als auch auf dem Drucker.

special character

*Sonderzeichen

Speicher; dynamischer

Ein dynamischer Speicher arbeitet im Prinzip wie ein Kondensator. Dieser Speichertyp muß dauernd wieder aufgeladen werden, damit die Information erhalten bleibt. Vorteil der dynamischen Speicher ist die relative Preisgünstigkeit und der niedrigere Stromverbrauch.

Speicher; statisch

Ein statischer Speicher arbeitet wie ein Flip-Flop. Dieser Speichertyp benötigt keine Wiederauffrischung. Diese Speicher sind aber relativ teuer, dafür aber problemloser in der Handhabung.

Speicheradresse

Jeder Speicherplatz in einem Computer hat eine eindeutige Nummer, die sogenannte Speicheradresse. Anhand dieser Adresse kann der Mikroprozessor in jede Speicherstelle schreiben und von jeder lesen.

Von BASIC aus kann mit Hilfe des *POKE- und des *PEEK-Befehls auf jede Speicheradresse zugegriffen werden.

Speicherplatz

Stelle, an der Variable, Zahlen, Ziffer, Sonderzeichen u.s.w. abgelegt werden. Jeder Speicherplatz kann über seine *Speicheradresse angesprochen werden.

Spezifikation

Das Erstellen der Spezifikation ist der erste Schritt zur Programmerstellung. In der Spezifikation wird genau festgelegt, was das Programm überhaupt leisten soll und in welche großen Teile das Programm zerlegt werden kann.
Weiterhin wird in der Spezifikation festgelegt, durch welche *Masken die Eingabe unterstützt werden soll.
In diesem Teil werden die Masken genau beschrieben, ebenso wie die Fehlbedienungen, die der Benutzer machen kann und wie diese Fehlbedienungen durch Fehlermeldungen quittiert werden. Bei *Dialogprogrammen sollte auch die Dialogstrategie beschrieben werden.
Die Spezfikation sollte auch noch eine genaue Beschreibung der ausgegebenen Listen beinhalten. Schließlich gehört in eine Spezifikation noch eine Beschreibung der zu verarbeitenden *Dateien.
Die Erstellung eines Programmes besteht aus den drei großen Schritten Spezifikation, *Entwurf und *Programmierung.

Sprites

Sprites sind programmierbare Figuren, die sich besonders schnell und einfach bewegen lassen.
Sprites bestehen beim Commodore 64 aus einer Matrix von 21*24 Punkten in beliebiger Farbe. Bei einer Auflösung von 21*12 Punkten sind bis zu vier verschiedene Farben in einer Figur möglich.

Anhand spezieller Register lassen sich Kollisionen der 8 möglichen Sprites untereinander und mit Hintergrundzeichen erkennen.

Bis zu 8 Sprites können gleichzeitig auf dem Bildschirm dargestellt und bewegt werden.

Sprites; Bewegung von

Zu jedem der 8 möglichen Sprites gibt es zwei Register, die die Position in X- und Y-Richtung angeben. Folgende Tabelle gibt Auskunft, welche Speicherstelle für welches Sprite relevant ist.

Speicherstelle	Sprite
53248	0/X-Richtung
53249	0/Y-Richtung
53250	1/X-Richtung
53251	1/Y-Richtung
53252	2/X-Richtung
53253	2/Y-Richtung
53254	3/X-Richtung
53255	3/Y-Richtung
53256	4/X-Richtung
53257	4/Y-Richtung
53258	5/X-Richtung
53259	5/Y-Richtung
53260	6/X-Richtung
53261	6/Y-Richtung
53262	7/X-Richtung
53263	7/Y-Richtung

Beispiel:

Das folgende kleine Programm soll das Prinzip verdeutlichen, nachdem Sprites bewegt werden.

```
10 V=53248
20 POKE V+21,1
30 POKE 2040,13
40 POKE V+39,0
50 FOR I=832 TO 894
60 POKE I,5
70 NEXTI
80 Y=200
90 FOR X=30 TO 229
100 Y=Y-1
110 POKE V,X
120 POKE V+1,Y
130 NEXTX
```

In Zeile 10 wird die *Variable V auf den Start der Register des Videochips gesetzt. Ab Speicherstelle 53248 sind die Register des Videochips gespeichert.

In Zeile 20 wird das Sprite 0 eingeschaltet.

Zeile 30 legt fest, aus welchem Speicherbereich die Informationen über das Aussehen des Sprites gelesen werden. Die Speicheradresse errechnet sich aus 13*64=832.

Zeile 40 bestimmt die Farbe des Sprites.

Die Zeilen 50 bis 70 definieren das Sprite. Näheres dazu in *Sprites, Erstellen von.

Die Zeilen 110 und 120 sind die eigentlichen Zeilen, die die Bewegung des Sprites bestimmen. Die Speicherstelle 53248 (=V) enthält nach obiger Tabelle die X-Position des Sprites.

Wird der Wert dieser Speicherstelle verändert, so ändert sich automatisch sofort die Position des Sprites auf dem Bildschirm entsprechend. Entsprechendes gilt für die Speicherstelle 53249, die die Y-Position enthält.

Durch eine einfache FOR-NEXT-Schleife läßt sich so ein Sprite sehr schnell über den Bildschirm bewegen.

Sprites; Erstellen von

Sprites werden aus einer Matrix von 24*21 Punkten gebildet. Die 21 Reihen enthalten somit je 3*8 Punkte. Daraus ergibt sich, daß zur Speicherung eines Sprites 21*3=63 Bytes notwendig sind.

Jeder dieser Punkte innerhalb eines Bytes hat nun eine Wertigkeit. Diese sind von links nach rechts die Zahlen 128, 64, 32, 16, 8, 4, 2 und 1. Wird ein Punkt gesetzt, so bekommt das entsprechende Byte diese Wertigkeit zugeordnet. Enthält das Byte bereits eine Zahl, so wird die Wertigkeit addiert.

Hat man so alle 63 Werte errechnet, kann man diese in einem beliebigen Speicherbereich innerhalb der unteren 16 Kilobyte abspeichern. Sie müssen allerdings darauf achten, daß der benutzte Bereich durch nichts anderes belegt ist.

Damit das Videochip weiß, wo Sie die Information nun gespeichert haben, müssen Sie noch einen Zeiger setzen. Dies sind die Speicherstellen 2040 - 2047 für die 8 Bytes. Der Inhalt dieser Speicherstelle, multipliziert mit 64, ergibt die Adresse, an der die Informationen über das Aussehen des Sprites gespeichert sind.

Enthält die Speicherstelle 2040 z.B. den Wert 13, so bedeutet dies, daß das Aussehen des Sprites mit der Nummer 0

von den Speicherstellen 832 bis 832+63=895 bestimmt wird.

Und so sieht eine solche Matrix aus:

```
-----------------------------------------------------
I I I I I I I I I I I I IxI I I I I I I I I I I I  0,8,0
-----------------------------------------------------
I I I I I I I I I I I IxIxIxI I I I I I I I I I I  0,28,0
-----------------------------------------------------
I I I I I I I I I I IxIxIxIxIxI I I I I I I I I I  0,60,0
-----------------------------------------------------
I I I I I I I I I IxIxIxIxIxIxIxI I I I I I I I I  0,126,0
-----------------------------------------------------
I I I I I I I I IxIxIxIxIxIxIxIxIxI I I I I I I I  0,255,128
-----------------------------------------------------
I I I I I I I I IxIxIxIxIxIxIxIxIxI I I I I I I I  0,255,128
-----------------------------------------------------
I I I I I I I I IxIxI I IxI I IxIxI I I I I I I I  0,201,128
-----------------------------------------------------
I I I I I I I I IxIxI I IxI I IxIxI I I I I I I I  0,201,128
-----------------------------------------------------
I I I I I I I I IxIxI I IxI I IxIxI I I I I I I I  0,201,128
-----------------------------------------------------
I I I I I I I I IxIxI I IxI I IxIxI I I I I I I I  0,201,128
-----------------------------------------------------
I I I I I I I I IxIxI I IxI I IxIxI I I I I I I I  0,201,128
-----------------------------------------------------
I I I I I I I I IxIxIxIxIxIxIxIxIxI I I I I I I I  0,255,1
-----------------------------------------------------
I I I I I I I I IxIxIxIxIxIxIxIxIxI I I I I I I I  0,255,1
-----------------------------------------------------
I I I I I I I I IxIxIxIxIxIxIxIxIxI I I I I I I I  0,255,1
-----------------------------------------------------
I I I I I I I I I IxIxI I I IxIxI I I I I I I I I  0,99,0
-----------------------------------------------------
I I I I I I I I I IxIxI I I IxIxI I I I I I I I I  0,99,0
-----------------------------------------------------
I I I I I I I I I IxIxI I I IxIxI I I I I I I I I  0,99,0
-----------------------------------------------------
I I I I I I I I IxIxIxIxI IxIxIxIxI I I I I I I I  0,247,128
-----------------------------------------------------
```

```
I I I I I I I I IxIxIxIxI IxIxIxIxI I I I I I I I  0,247,128
-----------------------------------------------
I I I I I I I I IxIxIxIxI IxIxIxIxI I I I I I I I  0,247,128
-----------------------------------------------
I I I I I I I I IxIxIxIxI IxIxIxIxI I I I I I I I  0,247,128
-----------------------------------------------
```

Neben den Zeilen stehen die Werte, die in den Bereich ab 832 gespeichert werden müssen.

Sprung

Durch einen Sprung wird der normale Programmablauf, der aufsteigend nach den Zeilennummern abläuft, geändert.
Ein Sprung setzt den Zeiger auf die nächste abzuarbeitende Zeile auf die Zeilennummer, die durch den Sprungbefehl angegeben ist.
Sprungbefehle sind z.B. GOTO oder GOSUB

Sprung; bedingter

Bedingte Sprünge werden durch den *IF-THEN-Befehl oder durch *ON-GOTO bzw. ON-GOSUB eingeleitet. Sie dienen dazu, den Programmablauf entsprechend den Eingaben des Benutzers zu beeinflussen.

Beispiel:

```
100 IF A=1 THEN 2000
110 REM Hier geht es normal weiter
.
.
.
2000 REM Hier geht es bei Erfüllung der Bedingung weiter
```

Sprung; unbedingter

Der unbedingte Sprung (GOTO) wird immer ausgeführt, er ist von keiner Bedingung abhängig.

Beispiel:

```
10 GOTO 10000
20 REM Hier stehen alle Unterprogramme
.
.
.
10000 REM Hier steht das Hauptprogramm
```

Sprungadresse

Die Sprungadresse ist in BASIC die Zeilennummer, zu der verzweigt wird. In anderen Programmiersprachen sind oft auch sogenannte symbolische Sprungadressen oder *Labels erlaubt. Diese sind unabhängig von Zeilennummern.
In SIMONS-BASIC können solche Labels mit Hilfe des PROC-Befehls definiert und mit dem CALL-Befehl angesprungen werden.

Spur

Jede *Floppy ist, damit diese verwaltet werden kann, in konzentrische Kreise, die sogenannten Spuren, eingeteilt. Die Anzahl der Spuren ist vom Diskettenlaufwerk abhängig.
Die Floppy des COMMODORE 64, die 1541, besitzt 35 Spuren. Mit Hilfe des *B-R- und des B-W-Befehls kann auf einzelne Spuren und Blöcke zugegriffen werden.

ST

Statusvariable; zeigt Fehler bei Ein Ausgabeoperationen an.

Sie dient dazu, Fehler zu entdecken, ohne den Programmablauf stoppen zu müssen.

Diese Variable kann nicht anderweitig für Rechenoperationen verwendet werden. Ansonsten läßt sie sich wie jede andere abfragen und ausgeben.

Diese Werte kann ST annehmen:

Wert	Bedeutung
0	alles ok
64	Dateiende
-128	Device not present

Mit Hilfe dieser Funktion läßt sich z.B. feststellen, wann eine sequentielle Datei fertig gelesen ist.

Beispiel:

```
10 OPEN1,8,2,"TEST,S,R"
20 INPUT#1,A$
30 IFST=64THENEND
40 PRINTA$
50 GOTO 20
```

In Zeile 30 wird die Variable ST abgefragt. Enthält diese den Wert 64, so ist die Datei zu Ende.

Stack

Der Stack ist ein schneller Zwischenspeicher. Wenn Sie in BASIC programmieren, haben Sie mit diesem Speicher nicht viel zu tun.
Lediglich müssen Sie beachten, daß bei GOSUB-Befehlen und FOR-NEXT-Schleifen immer eine bestimmte Anzahl von Bytes auf dem Stack abgespeichert. Werden Unterprogramme nicht ordnungsgemäß mit dem RETURN-Befehl verlassen oder FOR-NEXT-Schleifen einfach abgebrochen, so bleiben diese Bytes auf dem Stack stehen. Da die Speicherkapazität des Stacks beschränkt ist, kommt es so irgendwann zu einem Überlauf.
Der Rechner meldet sich dann mit der Fehlermeldung OUT-OF-MEMORY-ERROR. Dies erscheint, obwohl Sie dann mit der FRE-Funktion noch genügend freie Speicherplätze angezeigt bekommen. Der einzige Grund ist der Überlauf des Stacks.

Standardabweichung

Statistisches Maß um die Streuung der Meßwerte innerhalb einer Stichprobe zu ermitteln. Die Standardabweichung stellt die durchschnittliche Differenz der Meßwerte zu dem festgestellten *Mittelwert dar.
Ist die Standardabweichung groß, so bedeutet dies, daß die Meßwerte in einem großen Bereich liegen. Ist sie klein, liegen Sie in einem engbegrenzten Intervall.

stand by

Betriebsbereitschaft, der Computer wartet auf eine Eingabe.

Stapelverarbeitung

Dieser Begriff stammt aus der Groß-EDV. Man spricht immer dann von Stapelverarbeitung, wenn eine Reihe von Befehlen zuerst eingegeben, gespeichert und dann verarbeitet werden. Bei Großanlagen wird dieses Verfahren benutzt um die Rechenzeit der Zentraleinheit besser auszunutzen, da die Eingabe immer langsamer als die Verarbeitung ist. So besteht die Möglichkeit, daß mehrere Benutzer an *Terminals Daten erfassen und diese dann nach Fertigstellung abgearbeitet werden.

Die zweite Bedeutung erfährt dieser Begriff bei der Abarbeitung von Programmen. Programme werden bei großen EDV-Anlagen zumeist in Warteschlangen eingereiht und nacheinander abgearbeitet. Stapelverarbeitung wird auch als Batch-Betrieb bezeichnet.

Statement

Befehl

Jeder Befehl einer Programmiersprache wird als Statement bezeichnet, z.B. *GOTO oder *REM

Statistik; deskriptive

beschreibende Statistik; sie versucht die Zusammenhänge zwischen Daten zu ermitteln.
Typische Funktionen der deskriptiven Statistik sind Mittelwert, Varianz, Standardabweichung und der Korrelationskoeffizient.
Durch die deskriptive Statistik wird nur versucht,

Zusammenhänge zwischen Daten zu ermitteln und nicht Schlüsse auf die Zukunft gezogen.

Statistik; induktive

ermittelnde Statistik; Wahrscheinlichkeitsrechnung

Diese Form der Statistik versucht, den Zusammenhang zwischen den gegenwärtig gemessenen Ausprägungen der Merkmale und deren zukünftigen Werten zu ermitteln.
Wichtige Mittel der induktiven Statistik sind die Berechnung von Häufigkeitsverteilung, Hochrechnungen und verschiedene Testverfahren.
Die Testverfahren erlauben nachzuprüfen, wie groß die Wahrscheinlichkeit ist, daß die errechneten Trends zutreffen.

status

Zustand

status bit

Zustandsbit

status register

Statusregister, *ST

Steckmodul

Ein Steckmodul besteht aus einer Platine und einem *EPROM. Steckmodule werden in den COMMODORE 64 rechts hinten in den Modulschacht eingesteckt.
Diese Module enthalten meist Spiele, selten auch kommerzielle Programme.
Der große Vorteil der Steckmodule ist, daß die Ladezeiten von Disketten entfallen. Man braucht diese nur einzustecken, den Rechner einzuschalten und automatisch befindet man sich in diesem Programm.
Steckmodule benötigen meist keinen Platz vom Arbeitsspeicher des Rechners. Es sind auch Programmiersprachen auf Steckmodul erhältlich.

STOP

1.) Taste auf der Tastatur des COMMODORE 64.

2.) Befehl zur Unterbrechung des Programmablaufes

zu 1.) Mit dieser Taste ist es möglich, den Programmablauf eines BASIC-Programmes von außen her jederzeit zu stoppen. Diese Taste ist mit Hilfe des Befehls POKE788,52 abschaltbar. Dann hat diese Taste keinerlei Wirkung mehr.
Wird diese Taste zusammen mit der SHIFT-Taste betätigt, so wird ein Programm von Kassette geladen und automatisch gestartet. Der Tastaturcode dieser Taste ist 2.

zu 2.) Mit diesem Befehl ist es möglich, den Programmablauf zu unterbrechen, ohne daß die Variableninhalte verloren gehen.

SYNTAX:

STOP

So läßt sich auf einfachste Weise die korrekte Arbeit eines Programmes anhand von Zwischenergebnissen überprüfen.
Nach Überprüfung der Variableninhalte läßt sich das Programm mit Hilfe des CONT-Befehl weiterführen.

Stopp-Bit

Das Stopp-Bit wird bei der seriellen Datenübertragung verwendet. Es kennzeichnet das Ende der Übertragung eines *Bytes.
Verwendet werden üblicherweise 1, 1 1/2 oder 2 Stoppbits. Natürlich müssen Sender und Empfänger sich einig über diese Anzahl sein, da es sonst zu Datenfehlern kommt, d.h. die gesendeten Daten werden vom Empfänger falsch verstanden und verarbeitet.

storage

*Speicher

storage address

*Speicheradresse

storage capacity

*Speicherkapazität

String

Ein String ist eine Kette von Zeichen. Alle BASIC-Befehle, die mit dem $-Zeichen enden, dienen zur Manipulation von Strings (z.B. MID$,LEFT$ u.s.w.)
Strings werden zur Speicherung jeder Art von Daten verwendet. Zahlen lassen sich mit Hilfe des STR$-Befehls in Strings umwandeln. Dies ist besonders dann sinnvoll, wenn Zahlen einem bestimmten Format gedruckt oder gespeichert werden sollen.
Mit Hilfe des *VAL-Befehls ist es möglich, Strings wieder in Zahlen zurückzuverwandeln

STRING$

Befehl des Microsoft-BASIC zur Vervielfachung eines Strings.

SYNTAX:

STRING$(String,Wert)

Mit diesem Befehl läßt sich eine beliebige Zeichenkette beliebig oft vervielfachen, solange die maximal erlaubte Stringlänge von 255 nicht überschritten wird.

Beispiel:

```
10 INPUT "ZEICHENKETTE";Z$
20 INPUT "ANZAHL";A
30 A$=STRING$(Z$,A)
40 PRINTA$
50 GOTO10
```

Dies ergibt bei Eingabe von "AB" und 10:

ABABABABABABABABAB

Seite 226

Im SIMONS-BASIC zum COMMODORE 64 ist dieser Befehl ebenfalls vorhanden. Im normalen BASIC läßt er sich recht einfach simulieren:

Beispiel:

```
10 A$=""
20 INPUT "ZEICHENKETTE";Z$
30 INPUT "ANZAHL";A
40 FOR I=1 TO A
50 A$=A$ + Z$
60 NEXTI
70 PRINTA$
80 GOTO10
```

Mit Hilfe einer einfachen FOR-NEXT-Schleife in den Zeilen 40 bis 60 läßt sich der Befehl simulieren. Nachteil ist aber, daß dies nur recht langsam geht.

STRUKTO 64

Programmiersprache für den Commodore 64, die besonders die *strukturierte Programmierung unterstützt.
STRUKTO 64 ist eine Interpretersprache, die die Vorzüge von PASCAL und BASIC vereinigt. Leichte Erlernbarkeit, einfache Bedienung und ein eingebautes Toolkit zum Verbessern der Programme zeichnen diese Programmiersprache aus.
So wird auch die Erzeugung von Graphiken, die Erstellung von Sprites und die Erzeugung von Musik mit einfachen aber leistungsfähigen Befehlen unterstützt. Das Abspielen von Musik ist sogar unabhängig vom Programmablauf möglich.

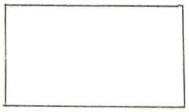
Struktogramm

Hilfsmittel zum Programmentwurf

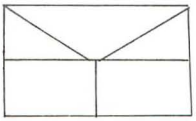
Struktogramme dienen der graphischen Darstellung eines Programmablaufes. Sie unterstützen besonders die strukturierte Programmierung, bei der möglichst keine Sprungbefehle verwendet werden.
Struktogramme sind programmiersprachenunabhängig. Besonders gut lassen sie sich aber in strukturierte Sprachen wie *PASCAL oder *STRUKTO 64 umsetzen.

Ein Struktogramm ist wie ein großes Rechteck aufgebaut, das viele kleine Rechtecke enthält. Die Ablaufrichtung ist von oben nach unten. Struktogramm bestehen hauptsächlich aus fünf verschiedenen Komponenten.

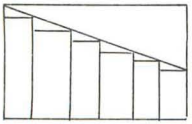
Operation:



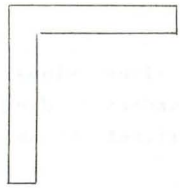
Verzweigung; einfach



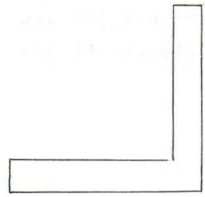
Verzweigung; mehrfach



Schleife, bei der die Anzahl der Durchläufe vor Beginn bekannt ist.



Schleife, bei der die Abbruchbedingung am Ende abgfragt wird.



subprogram

*Unterprogramm

SWAP

Befehl des Microsoft-BASIC um die Inhalte zweier Variablen zu vertauschen.

SYNTAX:

SWAP Variable,Variable

Mit diesem Befehl ist es möglich, die Inhalte zweier beliebiger Variablen zu vertauschen. Es spielt keine Rolle, ob es numerische oder Stringvariablen sind, allerdings müssen beide Variablen vom gleichen Typ sein.

Im BASIC des COMMODORE 64 läßt sich dieser Befehl folgendermaßen simulieren:

Beispiel:

```
10 Z$=A$
20 A$=B$
30 B$=Z$
```

Nachteil dieser Methode ist natürlich, daß ein Zwischenspeicher benötigt wird, hier die Hilfsvariable Z$.

Symbol

Sinnbild, Bild

Ein gutes Beispiel für Symbole sind die ASCII-Codes für die Buchstaben. Jeder Buchstabe wird durch eine Ziffernkombination symboliesiert.
So wird der Buchstabe a zum Beispiel durch die zwei Ziffern 97 dargestellt.

symbol address

symbolische Adresse, *Label

symbolic instruction code

frei übersetzt: Programmiersprache

Dieser Begriff erklärt sich daraus, daß anstatt der Zahlen, die der Computer nur versteht, Zeichen (Symbole) verwendet werden. Jedem Befehl einer Programmiersprache ist eine Zahl bzw. ein Code zugeordnet, mit dem dieser Befehl verschlüsselt wird. Nur so ist der Computer in der Lage, die Programmiersprache, die doch der menschlichen Sprache recht verwandt ist, zu verstehen.

sync character

Synchronisationszeichen, wird bei der Datenübertragung benötigt, damit Sender und Empfänger den Anfang der Übertragung gleich erkennen.

Syntax

Die Syntax eines Befehls gibt an, wie der Befehl richtig geschrieben wird und wie die Variablen einzusetzen sind. Wenn ein Befehl fallsch eingegeben wird, so meldet das Programm sich beim Ablauf mit *SYNTAX-ERROR.

Beispiel:

richtig	falsch
PRINT	PRUNT

Synthesizer

Gerät zur rein elektronischen Erzeugung von Klängen.
Im COMMODORE 64 befindet sich ein IC, das einen kompletten Synthesizer enthält. Dieser ist in der Lage, Rauschen, Musik und sogar menschliche Sprache zu erzeugen.
Es ist durchaus möglich, mit mehr oder weniger großem Aufwand alle Geräusche vom Gewehrschuß bis zum "Tür zu schlagen" oder "Schritten im Hausflur" zu erzeugen. So kann der COMMODORE 64 zur Untermalung von Dia-Vorträgen oder Filmen benutzt werden.

SYS

Befehl des BASIC des Commodore 64 um Maschinenprogramme aufzurufen

SYNTAX:

SYS Adresse

Mit diesem Befehl ist eine Verzweigung zu Maschinenspracheprogrammen möglich. Werden diese Programme mit einem RETURN-Befehl abgeschlossen, so wird nach

Abarbeitung des Maschinenprogramms der nächste Befehl im BASIC-Programm abgearbeitet.

Dieser Befehl findet besonders bei zeitkritischen Programmteilen Anwendung, deren Realisierung in BASIC zu langsam wäre. Es lassen sich auch Parameter an diese Unterprogramme übergeben (s. 64 Tips + Tricks)

System

Ein System ist eine sinnvolle Verknüpfung verschiedener Einzelkomponenten. In der Mathematik Systeme zur Lösung von Gleichung.

Unter einem Computersystem versteht man zumeist eine Anlage bestehend aus Zentraleinheit, Monitor, Speichermedium und Ausgabegerät (Drucker).

Systementwicklung

auch Systemengineering genannt

Zur Systementwicklung gehören alle Vorgänge bzw. Stufen der Programmentwicklung. Dazu gehören haupsächlich Analyse und Planung

Systemprogrammierung

Diese Art der Programmierung ist sehr hardwaremäßig orientiert. Von Systemprogrammierern werden *Betriebssysteme und *Utilities entwickelt.
Aus diesem Grund müssen diese Programmierer gute Kenntnisse

der Hardware besitzen.
Die Programmierung erfolgt fast immer in *Maschinensprache.

SX 64

Name der tragbaren Version des COMMODORE 64. Diese Konfiguration besteht aus dem COMmODORE 64, einer Floppy 1541 und einem kleinen Farbmonitor. Alle diese Geräte sind in einem Koffergehäuse mit Tragegriff untergebracht.

T

Taste der Tastatur des COMMODORE 64. Der *Tastaturcode dieser Taste ist 22. Außer dem Buchstaben T enthält sie folgende weitere Zeichen: t, ▀ und ▌ . Diese Zeichen haben folgende *ASCII-Codes:

	Großschrift/Graphik	Groß-/Kleinschrift
T	84	212
t	-	84
▀	163	-
▌	103	-

Diese Zeichen haben die folgenden *Bildschirmcodes:

	Großschrift/Graphik	Groß-/Kleinschrift
T	20	84
t	-	20
▀	99	-
▌	84	-

Die Umschaltung zwischen den beiden Modi kann durch gleichzeitige Betätigung der Tasten *SHIFT und *COMMODORE herbeigeführt werden.

TAB

BASIC-Befehl zur Positionierung der Druckposition, der in fast allen BASIC-Dialekten enthalten ist.

SYNTAX:

PRINT TAB(Wert);Variable

Dieser Befehl bewirkt, daß der Cursor zur Position "Wert" auf dem Bildschirm bewegt wird. Befindet sich der Cursor bereits hinter dieser Position, so wandert der Cursor in die nächste Zeile. Der TAB-Befehl kann nur in Zusammenhang mit dem PRINT-Befehl benutzt werden.

Beispiel:

```
10 PRINTTAB(10);"TEST";TAB(20);"TEST"
```

Ergebnis:

```
          TEST        TEST
```

Taktfrequenz

Die Taktfrequenz eines Mikroprozessors gibt an, mit welcher Geschwindigkeit dieser arbeitet. Der 6510-Mikroprozessor im COMMODORE 64 arbeitet mit ca. 980 kHz.
Im Allgemeinen gilt, je höher die Taktfrequenz, desto schneller der Prozessor. Doch hängt die tatsächliche Geschwindigkeit der Abarbeitung eines Befehls auch noch davon ab, inwieweit der Prozessor durch intelligente Coprozessoren von Routineaufgaben entlastet wird.

Tape

Band

An die meisten Homecomputer lassen sich ganz normale Kassettenrecorder als preiswerte Möglichkeit der Datenspeicherung anschließen. Der COMMODORE 64 bietet dazu die DATASETTE an. Diese arbeitet mit gewöhnlichen Musikkassetten zur Datenspeicherung. Diese Magnetbänder

werden auch als Tape bezeichnet.
Hauptsächlich werden darunter bisher die Magnetbänder der Computer der Mittleren Datentechnik (MDT) und der Großcomputer verstanden.

target language

Zielsprache, *Object code

Tastatur

Die Tastatur ist der Teil des Computers, der die Tasten zur Bedienung des Gerätes enthält.
Während man früher auf Grund der Größe der Bauteile gezwungen war, Tastatur und Computer zu trennen, schafft man es heute, Tastatur und Computer in einem Gehäuse unterzubringen.
Auch der COMMODORE 64 hat diesen praktischen und platzsparenden Aufbau.

Tastaturabfrage

Um festzustellen, welche Taste gedrückt worden ist, gibt es eine Speicherstelle in der *Zeropage des COMMODORE 64, die für jede Taste einen bestimmten Code enthält. Dies ist die Speicherstelle 203, die 64 enthält, wenn keine Taste gedrückt wird. Mittels eines kleinen Programms kann der Code für die Tasten abgefragt werden:

Beispiel:

```
10 GETA$
20 IF A$="" THEN 10
30 PRINTA$,PEEK(203)
40 GOTO 10
```

Dieses kleine Programm druckt den Buchstaben und den dazugehörigen Tastaturcode aus.
Die folgende Tabelle gibt an, zu welcher Taste welcher Code gehört:

Taste	Code
1	56
2	59
3	8
4	11
5	16
6	19
7	24
8	27
9	32
0	35
a	10
b	28
c	20
d	18
e	14
f	21
g	26
h	29
i	33
j	34
k	37
l	42
m	36
n	39
o	38
p	41
q	62

r	17
s	13
t	22
u	30
v	31
w	9
x	23
y	25
z	12
Pfeil links	57
+	40
-	43
Pfundtaste	48
HOME-Taste	51
DEL-Taste	0
Klammeraffe	46
*	49
Pfeil oben	54
RUN/STOP	2
:	45
;	50
=	53
RETURN	1
,	47
.	44
/	55
CRSR DOWN	7
CRSR RIGHT	2
Leertaste	60
F1	4
F3	5
F5	6
F7	3

Anhand der Speicherstelle 653 kann festgestellt werden, ob die SHIFT-, die CTRL- oder die COMMODORE-Taste gedrückt wurde.

Dabei wird die CTRL-Taste mit dem Wert 4 codiert, die COMMODORE-Taste mit 2 und die SHIFT-Taste mit 1.

Beispiel:

```
10 A=PEEK(653)
20 IF A=0 THEN 10
30 IF A=1 THEN PRINT"SHIFT" : GOTO 10
40 IF A=2 THEN PRINT"COMMODORE" : GOTO 10
50 IF A=4 THEN PRINT"CTRL" : GOTO 10
60 GOTO 10
```

Terminal

Als Terminal wird ein Bildschirmarbeitsplatz bezeichnet, der über eine Datenleitung an eine Computer angeschlossen ist. Das Terminal besitzt zumeist keine eigene Intelligenz. Es dient nur als Eingabe- und Ausgabestation.

Zumeist sind mehrere Terminals an einen großen Computer angeschlossen.

Testprogramm

Man unterscheidet zwei Arten von Testprogrammen. Diese dienen entweder zum Test der Hardware oder der Software.

Testprogramme zum Austesten der Hardware werden auch als Diagnoseprogramme bezeichnet. Diese testen z.B. den RAM-Speicher, die Ein-Ausgabebausteine, Farbe, Ton und so weiter.

Testprogramme für die Software versuchen alle denkbaren Anwendungsfälle zu simulieren. Dies geschieht mit Daten,

deren Ergebnis vorher von Hand bestimmt wurde. Stimmen Ergebnis des Programms und die von Hand errechneten überein, so arbeitet das Programm korrekt.

Textverarbeitung

Die wohl häufigste Verwendung finden Computer, besonders die Personal- oder Homecomputer in der Textverarbeitung.

Textverarbeitung bedeutet zuerst einmal, daß Texte nicht direkt auf dem Papier, sondern zuerst auf dem Bildschirm geschrieben werden. Dort können diese dann korrektur gelesen und verbessert werden. Anschließend lassen sie sich beliebig oft ausdrucken.

Damit ist aber nur ein kleiner Teil der Textverarbeitung beschrieben. Texte lassen sich auf dem Bildschirm in beliebiger Weise manipulieren. So ist es natürlich möglich, die Texte auf ein Speichermedium (meist Diskette) abzuspeichern und wieder einzulesen. Es ist ein leichtes, das Format eines Textes zu ändern, einzelene Passagen zu löschen, zu verschieben oder zu kopieren. Automatischen Zentrieren von Zeilen, Blocksatz, Einrückungen und so weiter gehören zu den Selbstverständlichkeiten einer guten Textverarbeitung.

Korrespondenz läßt sich mit Hilfe einer Textverarbeitung so standardisieren und der Arbeitsaufwand so erheblich verringern.
Rundschreiben, d.h. Standardbriefe mit variabler Adresse, lassen sich in kürzester Zeit zu Hunderten erstellen.

Textverarbeitung ist wohl die sinnvollste und rentabelste Verwendung für einen Computer im Büro.

Textverarbeitung, bis vor kurzem noch den teuren

Personalcomputern vorbehalten, ist jetzt auch auf dem COMMODORE 64 möglich. So wurde auch dieses Lexikon zum größten Teil mit dem COMMODORE 64 und der Textverarbeitung TEXTOMAT erstellt.

Thermodrucker

Thermodrucker arbeiten ohne Farbband, dafür aber mit einem speziellen Papier. Punkte auf dem Papier entstehen durch Erwärmung.
Thermodrucker, die zumeist als Matrixdrucker arbeiten, erwärmen durch ihre Nadeln das Papier. An der Stelle, an der die Nadel auf das Papier trifft, entsteht so ein Punkt.
Erheblicher Vorteil dieses Druckertypes ist seine sehr geringe Arbeitslautstärke. Nachteilig machen sich die geringe Geschwindigkeit und die hohen Kosten des Spezialpapiers bemerkbar.

TI

Variable, die den Zeitraum vom Einschalten des Rechners bis zum momentanen Zeitpunkt in Zehntelsekunden enthält.
Diese Variable steht nur auf COMMODORE-Computern zur Verfügung.

Beispiel:

```
10 TI$="000000"
20 FOR I=1 TO 10000
30 NEXTI
40 PRINT TI
```

Mit diesem Programm kann die Schnelligkeit des Computers überprüft werden. Eine leere FOR-NEXT-Schleife wird 10000mal

durchlaufen. Davor wird die Zeit auf 0 gestzt und anschließend die benötigte Zeit in Zehntelsekunden angezeigt.

TI$

Variable, die den Zeitraum vom Einschalten des Rechners bis zum momentanen Zeitpunkt in der Form Stunde, Minute, Sekunde enthält.
Dièse Variable kann beliebig gesetzt werden. Die Variable TI wird dadurch automatisch mitgesetzt.

Time-Sharing

Time-Sharing wird nur von größeren Computeranlagen betrieben. Dies bedeutet, daß mehrere Programme in einem Computer abgearbeitet werden.
Dies wird vom Rechner so verwaltet, daß er ein paar Sekunden das eine, die nächsten paar Sekunden das nächste Programm abarbeitet.
Der COMMODORE 64 besitzt durch seine Interruptsteuerung auch die Möglichkeit, mehrere Proramme gleichzeitig abzuarbeiten. Näheres dazu steht in dem Buch 64 Tips + Tricks.

Tintenstrahldrucker

Dieser Druckertyp arbeitet zumeist auch als Matrixdrucker. Er benötigt kein Farbband und auch kein Spezialpapier.
Die Punkte werden durch winzige Farbspritzer auf dem Papier erzeugt. Der Druckkopf enthält statt der üblichen Nadeln winzige Düsen, die die Farbe verspritzen.

Erheblicher Vorteil dieses Druckertypes ist seine sehr geringe Arbeitslautstärke.

Trace

Dieser Begriff steht für eine Betriebsweise, die auch Einzelschrittabarbeitung genannt wird.

In diesem Modus wird ein Programmschritt ausgeführt. Anschließend wird in BASIC die Programmzeile angegeben, die als nächstes angesprungen wird. In Maschinensprache wird der Inhalt des Programmzählers und die Inhalte der einzelnen Register angegeben.
Dann wird entweder automatisch oder auf Tastendruck der nächste Programmschritt ausgeführt.

Mit Hilfe dieses Modus ist es auf einfache Weise möglich, den logischen Ablauf und die Inhalte der Register bzw. Variablen zu überprüfen.
Auf dem COMMODORE 64 läßt sich dieser Modus nur mit Hilfe eines Maschinenspracheprogramms durchführen.

Treiber

Als Treiberprogramm werden zumeist solche Programme bezeichnet, die die Ausgabe von Daten auf Peripheriegeräte ermöglichen. Zumeist überwachen diese Programme den zeitlichen Ablauf der Ausgabe. Dabei gilt es zu beachten, daß z.B. einem Drucker erst das nächste Zeichen geschickt wird, wenn dieser zum Drucker auch bereit ist.
Außerdem führen solche Programme oft auch eine Codewandlung der Zeichen durch. So kann es bei Druckern, die nicht vom Hersteller des Computers stammen, durchaus vorkommen, daß ein Buchstabe im Computer anders codiert wird als auf dem

Drucker. Dann muß anhand einer Tabelle eine Umsetzung durchgeführt werden.

TROFF

Befehl des Microsoft-BASIC, der den *TRACE-Modus ausschaltet.

TRON

Befehl des Microsoft-BASIc, der den *TRACE-Modus einschaltet.

Typenraddrucker

Dieser Druckertyp arbeitet wie eine Schreibmaschine. Alle zur Verfügung stehenden Buchstaben sind fest auf einem Typenrad installiert. Bei Bedarf werden diese dann über ein Farbband gegen das Papier geschlagen, das über eine Gummiwalze gespannt ist.
Erheblicher Vorteil dieses Druckertypes ist das sehr gute Schriftbild. Nachteil ist die relativ langsame Geschwindigkeit dieser Drucker.
Viele elektronische Schreibmaschinen bieten heute auch die Möglichkeit, mit Hilfe einer geeigneten *Schnittstelle als Drucker für einen Computer zu fungieren.

U

Taste der Tastatur des COMMODORE 64. Der *Tastaturcode dieser Taste ist 30. Außer dem Buchstaben U enthält sie folgende weitere Zeichen: u, ▬ und ◪ . Diese Zeichen haben folgende *ASCII-Codes:

	Großschrift/Graphik	Groß-/Kleinschrift
U	85	213
u	-	85
▬	183	-
◪	121	-

Diese Zeichen haben die folgenden *Bildschirmcodes:

	Großschrift/Graphik	Groß-/Kleinschrift
U	21	85
u	-	21
▬	120	-
◪	85	-

Die Umschaltung zwischen den beiden Modi kann durch gleichzeitige Betätigung der Tasten *SHIFT und *COMMODORE herbeigeführt werden.

ultrviolet erasable PROM

UV-löschbares *EPROM

Underflow

Als Underflow wird eine Fehlermeldung beschrieben, die auftritt wenn eine Variable den kleinsten erlaubten Wert unteschreitet, der bei 2.93873587E-39 liegt.

unpack

entpacken, Daten werden bei der Speicherung auf Diskette zumeist gepackt, d.h. möglichst komprimiert. Der umgekehrte Vorgang, der beim Einlesen nötig wird, wird Entpacken genannt.

Unterprogramm

Charakteristisches Erkennungsmerkmal von Unterprogrammen ist, daß sie mit dem "RETURN"-Befehl abgeschlossen werden.

Unterprogramme werden für mehrfach in einem Programm auftretende Aufgaben verwendet. Diese Aufgaben werden einmal programmiert und als Unterprogramm abgelegt. Von beliebigen Programmteilen können diese dann aufgerufen werden.

Große Vorteile von Unterprogrammen sind, daß Sie Zeit bei der Entwicklung von Programmen sparen, da mehrfach auftretende Probleme nur einmal programmiert werden müssen. Zum anderen sparen sie aus dem gleichen Grund Speicherplatz für das Programm. Sehr wichtig ist auch, daß Unterprogramme die Übersichtlichkeit und Lesbarkeit eines Programmes entscheidend verbessern.

Beispiel:

In einem Programm müssen wiederholt prozentuelle Teile eines Betrages berechnet werden. Dabei ist der Betrag in der Variablen B und der Prozentsatz in der Variablen P zu übergeben. Das Ergebnis wird in der Variablen T gespeichert.

```
10 PRINT"Geben Sie den ersten Betrag und den ersten Zinssatz
ein"
20 INPUTB:INPUTP
30 GOSUB1000
40 PRINT"Der Teilbetrag lautet":T
.
.
.
999 END
1000 T=B*P/100
1010 RETURN
```

Dieses Beispiel ist natürlich nur sehr kurz, verdeutlicht aber die Vorgehensweise bei Unterprogrammen. Das Unterprogramm wird mit RETURN abgeschlossen und kann dann beliebig oft mit Hilfe des GOSUB-Befehls aufgerufen werden. Wichtig ist natürlich, daß immer alle Parameter versorgt werden.
Weiterhin ist zu beachten, daß das Hauptprogramm von dem ersten Unterprogramm mit *END abgeschlossen wird, bzw. daß sichergestellt wird, daß das Unterprogramm nicht abgearbeitet wird, ohne daß vorher ein *GOSUB-Befehl durchlaufen wurde. Wird dies nicht sichergestellt, so kann es zu dem Fehler "RETURN WITHOUT GOSUB ERROR" kommen.

Unterprogramme;geschachtelte

Wird in einem Unterprogramm ein weiteres Unterprogramm aufgerufen, so spricht man von geschachtelten Unterprogrammen.

Der COMMODORE 64 erlaubt es, bis zu einer Tiefe von 25 Unterprogramme aufzurufen. Allerdings nur, wenn keine Operationen ausgeführt werden, die Daten auf dem Stack ablegen (z.B. keine FOR-NEXT-Schleifen).

Update

Dieser englische Fachbegriff steht für die Erstellung der aktuellen Version einer Programmdiskette.
Bei Programmdisketten ist ein Update nur selten notwendig. Betroffen sind eigentlich nur kaufmännische oder steuerrechtliche Programme, die an Gesetzesänderungen angepaßt werden müssen. Außerdem kann es vorkommen, daß ein Programm erheblich verbessert wurde und der Kunde eine neue Version bekommt. Diese Art von Updates ist aber immer kostenpflichtig.

Nicht zu verwechseln ist dieser Begriff mit *Backup. Bei einem Backup wird nur eine Kopie der Diskette erstellt, während beim Update eine Diskette verändert wird.

Userport

Der Userport ist ein spezielle Schnittstelle der COMMODORE-Rechner. An diesem sind alle wichtigen Signale des Computers herausgeführt und stehen für die Kommunkikation mit der Außenwelt zur Verfügung.

Diese Schnittstelle findet besonders in der Meß- und Regeltechnik Anwendung. Doch auch Sie als Normalbenutzer können diese Schnittstelle z.B. als Centronics-Anschluß für einen Fremddrucker mit diesem Interface benutzen. Genaues finden Sie dazu in dem DATA BECKER Buch "64 Tips + Tricks"

Utilities

deutsch: Hilfsmittel

Als Utilities werden jede Art von Hilfsprogrammen bezeichnet. Dazu dienen z.B. Programme zur Neunummerierung von BASIC-Programmen, Programme zum Suchen und Ersetzen von Befehlen, Diskettenhilfsprogramme und so weiter.

V

Taste der Tastatur des COMMODORE 64. Der *Tastaturcode dieser Taste ist 31. Außer dem Buchstaben V enthält sie folgende weitere Zeichen: v, ◪ und ⊠ . Diese Zeichen haben folgende *ASCII-Codes:

	Großschrift/Graphik	Groß-/Kleinschrift
V	86	214
v	-	86
◪	190	-
⊠	118	-

Diese Zeichen haben die folgenden *Bildschirmcodes:

	Großschrift/Graphik	Groß-/Kleinschrift
V	22	86
v	-	22
◪	126	-
⊠	86	-

Die Umschaltung zwischen den beiden Modi kann durch gleichzeitige Betätigung der Tasten *SHIFT und *COMMODORE herbeigeführt werden.

V24

* RS232

VAL

Befehl des COMMODORE 64, der dazu dient, eine *String-Variable in eine numerische umzuwandeln.

SYNTAX :

VAL (Stringvariable oder Ausdruck)

Nehmen wir an, Sie wollen eine Eingabe programmieren, die genau 5 Stellen enthalten soll und nur numerisch sein darf. Die eingegebene Zahl muß dann rechnerisch weiterverarbeitet werden. Da der INPUT-Befehl weder eine Einschränkung der Stellenzahl erlaubt noch eine Einschränkung der erlaubten Zeichen und sowieso nicht sehr sicher ist (Cursortasten), müssen Sie die Eingabe mit Hilfe des GET-Befehls programmieren. Da der GET-Befehl aber nur mit Stringvariablen arbeitet, und Sie aber mit der Zahl weiterrechnen wollen, müssen Sie die Eingabe also in eine Zahl umformatieren. Das Problem läßt sich recht elegant lösen:

Beispiel:

```
1000 Z$=""
1010 FOR I=1 TO 5
1010 GETE$ : IF E$="" THEN 1010
1020 IF E$ "0" OR E$ "9" THEN 1010
1040 Z$=Z$+E$
1050 NEXTI
1060 Z=VAL(Z$)
```

Mit der Variablen Z können Sie nun weiterrechnen.

Variable

Eine Variable ist ein Stellvertreter für eine Zahl oder einen *String. Eine Variable ermöglicht es, eine Rechenoperation mehrfach mit verschiedenen Werten durchzuführen. Dazu werden der Variablen einfach verschiedene Werte zugewiesen. Durch Variablen ist es ebenfalls möglich, Strings zu verarbeiten und Stringoperationen mehrfach durchzuführen.

Erlaubte Variablennamen beim COMMODORE 64 sind:

A - Z
AA - ZZ
A1 - Z9
A$ - Z$
AA$ - ZZ$
A1$ - Z9$

Ausgenommen sind alle Namen, die den Anfangsbuchstaben von Befehlen entsprechen. So ist z.B. die Variable OR nicht erlaubt, da sie dem gleichnamigen BASIC-Befehl entspricht.

Mit Variablen kann ganz genauso wie mit konkreten Zahlen gerechnet werden. Das folgende Programm soll die Möglichkeit verdeutlichen, immer wieder den gleichen Rechenprozeß mit unterschiedlichen Zahlen durchzuführen, einfach indem den Variablen unterschiedliche Werte zugewiesen werden.

Beispiel:

```
10 INPUT A : INPUT B
20 C = A * B
30 PRINT"DAS ERGEBNIS VON"A" * "B" ist :"C
40 GOTO 10
```

Variable;global

Die Programmiersprache BASIC erlaubt nur sogenannte globale Variablen. Dies bedeutet, daß eine Variable für das gesamte Programm einschließlich aller Unterprogramme gültig ist. (siehe auch *Variable lokal)

Variable;lokal

Lokale Variablen gelten nur in einem bestimmten Programmteil. Dadurch ist es möglich, Unterprogramme zu schreiben, die gezielt nur ganz bestimmte Variablen des Hauptprogrammes beeinflussen. Ein versehentliches Benutzen der gleichen Variablennamen in Haupt- und Unterprogramm kann dadurch praktisch nicht mehr vorkommen.
Lokale Variablen sind eine wichtige Voraussetzung für die *strukturierte Programmierung, da nur so wirklich unabhängige Unterprogramme geschrieben werden können.

Variable;numerisch

Ist eine Variable numerisch, so kann sie nur als Platzhalter für Zahlen dienen. Numerische Variablen haben im BASIC des COMMODORE 64 die Namen:

A - Z
AA - ZZ
A1 - Z9

Werden auf eine numerische Variable Buchstaben oder Sonderzeichen zugewiesen, so kommt es zu der Fehlermeldung "TYPE MISMATCH ERROR".

Variablentypen

Das BASIC des COMMODORE 64 unterscheidet zwischen zwei Variablentypen, den *numrischen und den *alphanumerischen. Die alphanumerischen Variablen können alle im Rechner vorhandenen Zeichen, seien es Sonderzeichen, Zahlen oder Buchstaben verarbeiten.
In anderen Programmiersprachen existieren auch noch andere Typen. Sehr häufig sind auch noch die Boolschen Variablen anzutreffen. Diese können immer nur zwei Werte speichern, entweder ja oder nein. Der Vorteil dieser Variablen ist unter anderem, daß sie sehr speicherplatzsparend einzusetzen sind.

Variablentypen;Def von

In der Programmiersprache BASIC ist eine Definition von Variablentypen nicht notwendig. Anhand von Kennzeichen ($,%) erkennt BASIC die verschiedenen Variablentypen.

Andere Programmiersprachen wie zum Beispiel PASCAL verlangen, daß alle Variablen zu Beginn des Programms definiert werden müssen. Die Variablen werden im Programm dann nur noch anhand der Namen aufgerufen, sie enthalten kein Kennzeichen mehr über ihren Typ. Ebenfalls ist es möglich, eigene Variablentypen zu definieren, die zum Beispiel einen eingeschränkten Wertebereich haben.

In PASCAL sieht dies typisch so aus:

```
VAR  TIEFE : REAL
     ANZAHL : INTEGER
     WERT : 1..9
```

Die Variable TIEFE wird als Gleitkommazahl, die Variable

ANZAHL als Integerzahl behandelt. WERT kann nur Zahlen zwischen 1 und 9 enthalten. Wird im Programm versucht, diesen Variablen unerlaubte Werte zuzuweisen, so beschwert sich der Computer sofort.
Damit lassen sich recht einfach Bediener- und sogar logische Fehler erkennen.

VARPTR

Befehl des Microsoft-BASIC zur Ermittlung der Adresse der Speicherplätze, in denen der Wert einer Variable gespeichert ist.

SYNTAX:

VARPTR (Variablenname)

Bei numerischen Variablen ergibt diese Funktion den Speicherplatz, bei dem die Speicherung des Wertes beginnt.

Bei Stringvariablen ergibt diese Funktion die Adresse, bei der die Speicherung des sogenannten Stringdescriptors beginnt.
Dieser besteht beim COMMODORE 64 aus drei Byte: die beiden ersten enthalten die Adresse, bei der die Speicherung des Strings tatsächtlich beginnt, das dritte enthält die Länge des Strings.

Vergleichsoperationen

Diese spezielle Art von Rechenoperationen dient als Entscheidungsgrundlage für den Programmablauf. Der COMMODORE 64 kennt drei verschiedene Vergleichsoperationen: kleiner, größer und gleich.

Erlaubt sind außerdem noch Kombinationen wie:

- kleiner oder gleich
- größer oder gleich
- kleiner oder größer (entspricht ungleich)

Verwendet werden diese Operationen, um Entscheidungen fällen zu können. Ein typischer Fall sieht z.B. so aus :

Beispiel:

```
10 INPUT A$
20 IF A$="ENDE" THEN END
   .
   .
   .
```

In diesem Fall wird die Gleichheit überprüft und entsprechend im Programm fortgefahren.

Durch die Vergleichsbefehle ist es möglich, den zulässigen Wertebereich von numerischen Variablen einzuschränken. Weiterhin kann eine Variable auf eine Gruppe von zulässigen Werten überprüft werden.

VERIFY

Dieser BASIC-Befehl dient dazu, die korrekte Abspeicherung eines Programmes zu üüberprüfen

SYNTAX :

VERIFY "Programmname",0 oder 8

"Programmname" ist der Name des zu überprüfenden Programms. Dieser kann bei Kassettenbetrieb weggelassen werden.

Ebenfalls kann bei Kassettenbetrieb das Komma und die 0 weggelassen werden. Dagegen sind bei Floppybetrieb das Komma und die 8 zwingend notwendig.

Der COMMODORE 64 meldet sich nach Eingabe dieses Befehls zurück mit

```
SEARCHING FOR "Programmname"
VERIFYING
OK
```

Stellt der COMMODORE 64 einen Übertragungsfehler fest, so meldet er sich mit der Meldung "VERIFY ERROR". Dann empfiehlt sich die Verwendung einer anderen Diskette.

Verzögerungsschleife

Eine Verzögerungsschleife verlangsamt den Programmablauf. Dies kann sinnvoll sein, wenn z.B. Text auf dem Bildschirm erscheint und der Benutzer Zeit haben muß, diesen zu lesen. Typisch wird eine solche Schleife mit dem FOR-NEXT-Befehl aufgebaut:

```
1000 FOR W=1 TO 1000
1010 NEXTW
```

Am benutzerfreundlichsten ist es, in die Warteschleife eine Tastaturabrage zu integrieren. So kann der Leser des Textes auf Tastendruck weitermachen, andererseits geht es aber auch nach einer bestimmten Zeit automatisch weiter. Dies könnte so aussehen:

Beispiel:

```
1000 FOR W=1 TO 1000
```

```
1010 GETE$ : IF E$="" THEN NEXTW : GOTO 1040
1020 W=1000
1030 NEXTW
1040 REM HIER GEHT ES WEITER
```

Solange keine Taste gedrückt ist, wird die Verzögerungsschleife weiter gezählt. Sobald eine Taste gedrückt ist, wird die Schleife verlassen. Um den Stack nicht unnötig zu belasten, wird die Schleife dann noch ordnungsgemäß beendet.

Verzweigung

Eine Verzweigung dient immer dazu, den normalen Programmablauf, der aufsteigend den Zeilennummern nach abläuft, abzuändern. Es gilt zwischen einfachen und mehrfachen Verzweigungen zu unterscheiden.

Eine einfache Verzweigung ist der GOTO-Befehl. Er veranlaßt, daß das Programm an einer ganz bestimmten Zeilennummer fortgeführt wird. (*Sprung)

Eine mehrfache Verzweigung stellt der ON-GOTO-Befehl dar. Abhängig von dem Wert einer *Variablen wird das Programm an einer bestimmten Stelle fortgeführt. Typisch wird dieser Befehl bei einem Menue verwendet. Die einzelnen Punkte erhalten Nummern anhand derer sie angewählt werden können. Diese Nummern dienen dann gleichzeitig dem Programm als Entscheidungskriterium für die Verzweigung.

W

Taste der Tastatur des COMMODORE 64. Der *Tastaturcode dieser Taste ist 9. Außer dem Buchstaben W enthält sie folgende weitere Zeichen: w, ⊞ und ◯ . Diese Zeichen haben folgende *ASCII-Codes:

	Großschrift/Graphik	Groß-/Kleinschrift
W	87	215
w	-	87
⊞	179	-
◯	119	-

Diese Zeichen haben die folgenden *Bildschirmcodes:

	Großschrift/Graphik	Groß-/Kleinschrift
W	23	87
w	-	23
⊞	115	-
◯	87	-

Die Umschaltung zwischen den beiden Modi kann durch gleichzeitige Betätigung der Tasten *SHIFT und *COMMODORE herbeigeführt werden.

Wertzuweisung

Als Wertzuweisung kann im Prinzip jede Rechenoperation bezeichnet werden. Mit jeder Wertzuweisung erhält eine *Variable einen Inhalt, einen Wert. Der Operator für eine Wertzuweisung ist das Gleichheitszeichen "=". Typisch sieht dies so aus:

A$="TEST"

Oder es wird eine Rechnung ausgeführt:

B = B * 45

Diese Rechnung erscheint auf den ersten Blick unsinnig, mathematisch betrachtet ist das Gleichheitszeichen sogar falsch. Doch dies soll keine Gleichung im mathematische Sinne sein, sondern eine Zuweisung. Der Variablen links des Gleichheitszeichens wird der Wert der Rechnung rechts des Zeichens zugewiesen.

Eine Wertzuweisung ist immer so aufgebaut, daß das Ergebnis links und die Rechnung oder der zuzuweisende Wert rechts vom Gleichheitszeichen steht.

WHILE...WEND

Befehl des Microsoft-BASIC, der zum Aufbau von Schleifen dient, bei denen bei Beginn noch nicht feststeht wie oft sie durchlaufen werden.

SYNTAX:

WHILE Bedingung .... WEND

Ein kurzes Sortierprogramm soll die Verwendung des Befehls erläutern:

Beispiel:

In dem Array A$(I) sollen 15 Namen gespeichert sein, die sortiert werden müssen.

```
100 F=-1
110 WHILE F
120 F=0
130 FOR I = 1 TO 14
140 IF A$(I)< = A$(I+1) THEN 190
150 Z$ = A$(I)
160 A$(I) = A$(I+1)
170 A$(I+1) = Z$
180 F=-1
190 NEXTI
200 WEND
```

Simulieren läßt sich dieser Befehl im BASIC des COMMODORE 64 mit Hilfe der IF-THEN-Anweisung:

```
100 F=-1
110 IF F=0 THEN 210
120 F=0
130 FOR I = 1 TO 14
140 IF A$(I)< = A$(I+1) THEN 190
150 Z$ = A$(I)
160 A$(I) = A$(I+1)
170 A$(I+1) = Z$
180 F=-1
190 NEXTI
200 GOTO 110
210 REM Hier geht es weiter
```

WIDTH

Dieser Befehl legt die maximale Länge einer Ausgabezeile fest.

SYNTAX:

WIDTH Länge

Länge muß eine ganzzahlige Zahl sein. Nützlich ist dieser Befehl, wenn Papier verwendet wird, daß schmaler ist als die mögliche Druckbreite.

Beim COMMODORE 64 läßt sich dieser Befehl recht einfach mit Hilfe des LEFT$-Befehls simulieren.

Beispiel:

Jede Ausgabezeile wird dazu in die Variable Z$ überwiesen. Diese wird dann auf die maximal erlaubte Länge (L) gekürzt wenn nötig:

```
100 IF L<=LEN(Z$)THEN120
110 Z$=LEFT$(Z$,L)
120 REM
```

Genauso läßt sich auch die Ausgabe auf dem Bildschirm handhaben.

Winchester disk

Winchester-Platte, *Festplatte

window

Fenster, Bereich auf dem Bildschirm, in dem ohne Veränderung der anderen Bereiche gearbeitet werden kann.

Wissenschaftliche Notation

auch Potenzschreibweise genannt.

Die wissenschaftliche Notation findet Anwendung bei der Darstellung besonders großer oder besonders kleiner Zahlen.
Die Zahl 1230000000000000000
läßt sich so darstellen : 123E17

Der Wert errechnet sich aus
123 * 10 ^ 17

WRITE

Befehl der Microsoft-BASIC zur Ausgabe auf den Bildschirm

SYNTAX:

WRITE Liste

Dieser Befehl dient zur Ausgabe des Inhaltes von Variablen auf dem Bildschirm. Im Gegensatz zum PRINT-Befehl werden Kommata als Trennungszeichen mit ausgegeben und Zeichenketten in Anführungszeichen eingeschlossen.

word processing

*Textverarbeitung

X

Taste der Tastatur des COMMODORE 64. Der *Tastaturcode dieser Taste ist 23. Außer dem Buchstaben X enthält sie folgende weitere Zeichen: x, ◱ und ♣ . Diese Zeichen haben folgende *ASCII-Codes:

	Großschrift/Graphik	Groß-/Kleinschrift
X	88	216
x	-	88
◱	189	-
♣	120	-

Diese Zeichen haben die folgenden *Bildschirmcodes:

	Großschrift/Graphik	Groß-/Kleinschrift
X	24	88
x	-	24
◱	125	-
♣	88	-

Die Umschaltung zwischen den beiden Modi kann durch gleichzeitige Betätigung der Tasten *SHIFT und *COMMODORE herbeigeführt werden.

Y

Taste der Tastatur des COMMODORE 64. Der *Tastaturcode dieser Taste ist 25. Außer dem Buchstaben Y enthält sie folgende weitere Zeichen: y, ▬ und ▯ . Diese Zeichen haben folgende *ASCII-Codes:

	Großschrift/Graphik	Groß-/Kleinschrift
Y	89	217
y	-	89
▬	183	-
▯	104	-

Diese Zeichen haben die folgenden *Bildschirmcodes:

	Großschrift/Graphik	Groß-/Kleinschrift
Y	25	89
y	-	25
▬	119	-
▯	89	-

Die Umschaltung zwischen den beiden Modi kann durch gleichzeitige Betätigung der Tasten *SHIFT und *COMMODORE herbeigeführt werden.

Z

Taste der Tastatur des COMMODORE 64. Der *Tastaturcode dieser Taste ist 12. Außer dem Buchstaben Z enthält sie folgende weitere Zeichen: z, ▙ und ♦ . Diese Zeichen haben folgende *ASCII-Codes:

	Großschrift/Graphik	Groß-/Kleinschrift
Z	90	218
z	-	90
▙	173	-
♦	122	-

Diese Zeichen haben die folgenden *Bildschirmcodes:

	Großschrift/Graphik	Groß-/Kleinschrift
Z	26	90
z	-	26
▙	109	-
♦	90	-

Die Umschaltung zwischen den beiden Modi kann durch gleichzeitige Betätigung der Tasten *SHIFT und *COMMODORE herbeigeführt werden.

Z80

Der Z80 ist ein weiterer *Mikroprozessor. Er besitzt wie der im COMMODORE 64 eingebaute 6510 eine 8-Bit-Struktur, unterscheidet sich aber sonst erheblich von diesem. Über ein Steckmodul ist es möglich, einen Z80-Prozessor in den COMMODORE 64 einzusetzen. Mit diesem Modul wird dieser

*CP/M-fähig. Dadurch steht eine riesige Softwarepalette zur Verfügung.

Zahlensystem

Ein Zahlensystem ist gekennzeichnet durch die Basiszahl. Das üblich verwendete Zahlensystem ist das Dezimalsystem, dessen Basis die Zahl 10 ist.

Dies bedeutet, daß jede Ziffer mit einer Zehnerpotenz multipliziert werden muß, um den Wert der Zahl zu errechnen.

Beispiel:

345 errechnet sich aus 3*10↑2 + 4*10↑1 + 5*10↑0

In der Computerwelt finden das Binärsystem (Basis 2), das Oktalsystem (Basis 8) und das Hexadezimalsystem (Basis 16) Anwendung.

Denkbar sind Zahlensysteme mit jeder beliebigen Basis. Die sinnvolle Auswahl hängt lediglich vom verfolgten Zweck ab.

Zähler

Einer der wichtigsten Bestandteile eines Programms sind die Zähler. Diese werden z.B. in *Programmschleifen verwendet, um festzustellen, wann diese zu Ende ist. Bei jedem Durchgang wird der Zähler um eins erhöht. Hat er den Endwert erreicht, wird die Schleife abgebrochen.

```
10 Z=0
20 PRINT "HALLO"
30 Z=Z+1
```

```
40 IF Z=10 THEN END
50 GOTO 20
```

Das vorstehende Programm druckt zehnmal das Wort "HALLO" aus. Die *Variable Z fungiert hier als Zähler. Nach jedem Drucken wird sie um eins erhöht. Anschließend wird der Endwert abgefragt und entsprechend verzweigt.

Zähler werden auch benutzt, um bei der Ausgabe auf Drucker den Seitenumbruch festzustellen.

Beispiel:

```
1000 OPEN 1,4
1010 Z=3
1020 PRINT#1
1030 PRINT#1
1040 PRINT#1,Z
1050 Z = Z+1
1060 IF Z=70 THEN 1080
1070 GOTO 1040
1080 PRINT#1
1090 PRINT#1
1100 GOTO 1010
```

In diesem kleinen Programm werden am Anfang und Ende jeder Zeile zwei Leerzeilen gedruckt um die Perforierung zu überbrücken.

Zeichen

Als Zeichen wird jedes Symbol bezeichnet, daß sich durch einen Tastendruck auf dem Bildschirm hervorrufen läßt. Ein Zeichen wird beim COMMODORE 64 intern immer in einem *Byte verschlüsselt.

Zeichenkettenoperationen

Eine Reihe von BASIC-Befehlen sind mit dem $-Zeichen als letztem Buchstaben gekennzeichnet. Alle diese Befehle dienen der Zeichenkettenoperation. Sie werden hauptsächlich benutzt, um Buchstaben oder Worte zu manipulieren. So ist es möglich, einzelne Buchstaben zu ersetzen, Wörter miteinander zu verbinden oder zu trennen.

Zeichensatz

Als Zeichensatz wird die Menge aller dem Computer zur Verfügung stehenden *Zeichen definiert. Der COMMODORE 64 besitzt zwei Zeichensätze mit jeweils 256 Zeichen. Der Erste enthält alle Großbuchstaben und diverse Grafikzeichen, der zweite enthält alle Klein-, Großbuchstaben und einige Grafikzeichen.
Mit Hilfe der SHIFT- und der COMMODORE-Taste (links unten) kann zwischen den beiden Zeichensätzen hin- und hergeschaltet werden.
Softwaremäßig ist eine Umschaltung auf Groß/Grafikmodus mit dem Befehl PRINTCHR$(142), auf Klein/Großschrift mit PRINTCHR$(14) möglich.

Zeichenvorrat

Der Zeichenvorrat resultiert aus dem Zeichensatz. Da der COMMODORE zwei unterschiedliche Zeichensätze besitzt, besteht sein Zeichenvorrat aus erheblich mehr Zeichen als der der meisten anderen Computer.

Zeilennummer

In BASIC-Programmen wird jeder Befehl mit einer Zeilennummer versehen. Anhand dieser Nummer ist im Normalfall die Reihenfolge der Abarbeitung der Befehle zu erkennen. Sprünge im Programm und Unterprogrammaufrufe sind nur bezüglich einer Zeilennummer in BASIC möglich.
Die kleinste in BASIC mögliche Zeilennummer ist 0, die größte 63537

Zentraleinheit

*CPU

Zeropage

Die Zeropage ist eine Besonderheit alle Microcomputer, die mit dem Microrpzessor 6502 oder einem verwandten Typ arbeiten. Als Zeropage wird der Adressbereich von 0 bis 255 bezeichnet. Dieser Adressbereich kann vom Prozessor dank einer besonderen Adressierungstechnik mit 2-Byte-Befehlen angesprochen werden. Für den gesamten sonstigen Bereich werden immer 3 Byte benötigt. Dadurch kann mit diesen Speicherstellen sowohl platz- als auch zeitsparend gearbeitet werden.
Die Zeropage wird für die wichtigsten Variablen benutzt und

vor allem werden die Speicherstellen paarweise als Zeiger auf wichtige Adressen im übrigen Speicher genutzt.
So enthalten z.B. die Speicherstellen 43 und 44 die Zeiger auf den Anfang des BASIC-Programmspeichers.

Zufallsgenerator

Der Zufallsgenerator im COMMODORE 64 Basic erzeugt Zahlen ohne jegliches System zwischen 0 und 1. Um Zahlen in einem anderen Bereich zu erhalte, müssen diese Zahlen entsprechend rechnerisch behandelt werden. Das folgende Programm erzeugt zufällig 100 Buchstaben, deren ASCII-Code bekanntlich zwischen 65 und 90 liegt.

```
10 FOR I = 1 TO 100
20 B=RND(1)*25+65
30 PRINTCHR$(B)
40 NEXT I
50 END
```

Zufallszahl

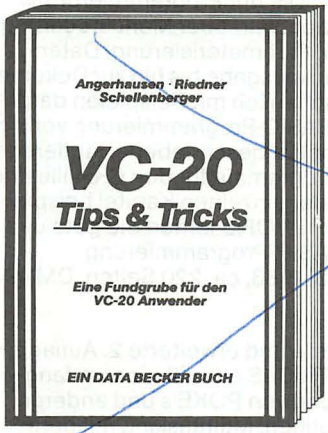
Als Zufallszahl wird eine von einem *Zufallsgenerator erzeugte Zahl bezeichnet.
In der Computerwelt bedient man sich dazu elektronischer Verfahren. Das bekannteste mechanische Verfahren zur Erzeugung von Zufallszahlen ist wohl die Ziehung der Lottozahlen.

Zwischenspeicher

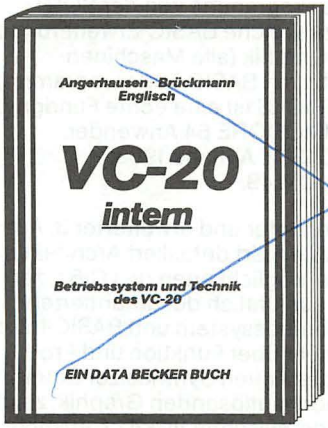
Als Zwischenspeicher wird jede *Variable bezeichnet, in der irgendein Ergebnis vorübergehend aufbewahrt wird.

Zwischenspeicher werden sehr häufig benutzt, um Unterprogrammen bestimmte Parameter zu übergeben und die Ergebnisse zurückzuerhalten.
Ebenfalls eine häufige Anwendung finden Zwischenspeicher, wenn der Inhalt von Variablen (*Sortieren) ausgetauscht werden muß.

DATA BECKER BÜCHER



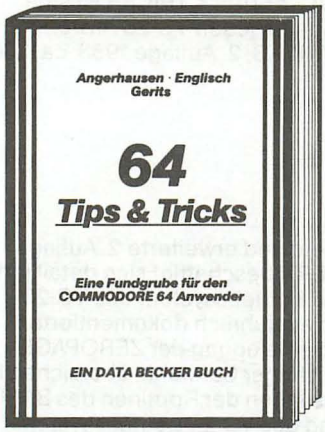
Die überarbeitete und erweiterte 2. Auflage von VC-20 TIPS & TRICKS enthält eine detaillierte Beschreibung der Programmierung von Sound und Graphik des VC-20, mehr über Speicherbelegung, Speichererweiterung und die optimale Nutzung der einzelnen Speichermodule, BASIC-Erweiterungen zum Eintippen, umfangreiche Sammlung von Poke's und anderen nützlichen Routinen, zahlreiche interessante Beispiel- und Anwendungsprogramme, komplett dokumentiert und fertig zum Eintippen (z. B. Spiele, Funktionenplotter, Graphik Editor, Sound Editor) und vieles andere mehr. VC-20 TIPS & TRICKS ist eine echte Fundgrube für jeden VC-20 Anwender. VC-20 TIPS & TRICKS, 2. Auflage 1983, ca. 230 Seiten, DM 49,–.



Die überarbeitete und erweiterte 2. Auflage von VC-20 INTERN beschäftigt sich detailliert mit Technik und Betriebssystem des VC-20 und enthält ein ausführlich dokumentiertes ROM-Listing, die Belegung der ZEROPAGE und anderer wichtiger Bereiche, übersichtliche Zusammenfassungen der Routinen des BASIC-Interpreters und des VC-20 Betriebssystems, eine Einführung in die Programmierung in Maschinensprache, eine detaillierte Beschreibung der Technik des VC-20 und als Clou drei Original COMMODORE Schaltpläne zum Ausklappen! Damit ist VC-20 INTERN für jeden interessant, der sich näher mit Technik und Maschinenprogrammierung des VC-20 auseinandersetzen möchte. VC-20 INTERN, 2. Auflage 1983, ca. 230 Seiten, DM 49,–.

DATA BECKER BÜCHER

Wer besser und leichter in BASIC programmieren möchte, der braucht dieses neue Buch. 64 FÜR PROFIS zeigt, wie man erfolgreich Anwendungsprobleme in BASIC löst und verrät Erfolgsgeheimnisse der Programmierprofis. Vom Programmentwurf über Menüsteuerung, Maskenaufbau, Parameterisierung, Datenzugriff und Druckausgabe bis hin zur Dokumentation wird anschaulich mit Beispielen dargelegt, wie gute BASIC-Programmierung vor sich geht. Fünf komplett beschriebene, lauffertige Anwendungsprogramme für den C-64 illustrieren den Inhalt der einzelnen Kapitel beispielhaft. Mit 64 FÜR PROFIS lernen Sie gute und erfolgreiche BASIC-Programmierung.
64 FÜR PROFIS, 1983, ca. 220 Seiten, DM 49,–.



Die überarbeitete und erweiterte 2. Auflage von 64 TIPS & TRICKS enthält eine umfangreiche Sammlung von POKE's und anderen nützlichen Routinen, Multitasking mit dem C-64, hochauflösende Graphik und Farbe für Fortgeschrittene, mehr über CP/M auf dem C-64, mehr über Anschluß- und Erweiterungsmöglichkeiten durch USER PORT und EXPANSION PORT, sowie zahlreiche ausführlich dokumentierte Programme von der SORT-Routine über zahlreiche BASIC-Erweiterungen bis hin zur 3D-Graphik (alle Maschinenprogramme jetzt mit BASIC-Ladeprogramm!). 64 TIPS UND TRICKS ist eine echte Fundgrube für jeden COMMODORE 64 Anwender.
64 TIPS & TRICKS, 2. Auflage 1983, ca. 290 Seiten, DM 49,–.

Jetzt in überarbeiteter und erweiterter 3. Auflage: 64 INTERN erklärt detailliert Architektur und technische Möglichkeiten des C-64, zerlegt mit einem ausführlich dokumentierten ROM-Listing Betriebssystem und BASIC-Interpreter, bringt mehr über Funktion und Programmierung des neuen Synthesizer Sound Chip und der hochauflösenden Graphik, zeigt die Unterschiede zwischen VC-20, C-64 und CBM 8000 und gibt Hinweise zur Umsetzung von Programmen. Zahlreiche lauffertige Beispielprogramme, Schaltbilder und als Clou: zwei ausführlich dokumentierte Original COMMODORE Schaltpläne zum Ausklappen. Dieses Buch sollte jeder 64-Anwender und Interessent haben. 64 INTERN, 3. Auflage 1983, ca. 320 Seiten, DM 69,–.

DATA BECKER BÜCHER

Das sollte Ihr erstes Buch zum COMMODORE 64 sein: 64 FÜR EINSTEIGER ist eine sehr leicht verständliche Einführung in Handhabung, Einsatz, Ausbaumöglichkeiten und Programmierung des COMMODORE 64, die keinerlei Vorkenntnisse voraussetzt. Schritt für Schritt führt das Buch Sie in die Programmiersprache BASIC ein, wobei Sie nach und nach eine komplette Adressenverwaltung erstellen, die Sie anschließend nutzen können. Zahlreiche Abbildungen und viele Anregungen zum sinnvollen Einsatz des COMMODORE 64.
Das Buch ist sowohl als Einführung als auch als Orientierung vor dem 64er Kauf gut geeignet. Ca. 200 Seiten, DM 29,–.

Graphik ist eine der Hauptstärken des COMMODORE 64. Mit diesem neuen Buch lernen Sie, wie Sie die graphischen Fähigkeiten programmtechnisch optimal nutzen. Der Inhalt reicht von den Grundlagen der Graphikprogrammierung über das Erzeugen einfacher Figuren, die Arbeit mit Sprites, Zeichensatzprogrammierung, Hardcopy und IRQ-Handhabung bis hin zur Funktionendarstellung, Laufschrift, Statistik, 3-D, CAD, den Geheimnissen der Actionsspiele und Lightpenanwendungen. Zahlreiche Beispielprogramme ergänzen dieses Buch, das die faszinierende Computertechnik jedermann zugänglich macht.
Ca. 250 Seiten, DM 39,–.

So etwas haben Sie gesucht:
Umfassendes Nachschlagewerk zum COMMODORE 64 und seiner Programmierung. Allgemeines Computerlexikon mit Fachwissen von A–Z und Fachwörterbuch mit Übersetzungen wichtiger englischer Fachbegriffe – das DATA BECKER LEXIKON ZUM COMMODORE 64 stellt praktisch drei Bücher in einem dar. Es enthält eine unglaubliche Vielfalt an Informationen und dient so zugleich als kompetentes Nachschlagewerk und als unentbehrliches Arbeitsmittel. Viele Abbildungen und Beispiele ergänzen den Text. Ein Muß für jeden COMMODORE 64 Anwender.
Ca. 350 Seiten, DM 49,–.

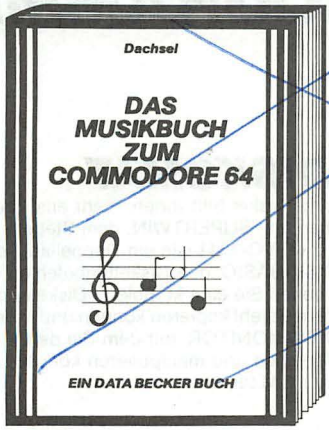
DATA BECKER BÜCHER

Eine leicht verständliche Einführung in die Programmierung des COMMODORE 64 in Maschinensprache und Assembler für alle diejenigen, denen die Programmierung in BASIC nicht mehr ausreicht. Beispiele erläutern jeden neuen Befehl. Zur komfortablen Eingabe und zum Austesten Ihrer Maschinenprogramme enthält das Buch einen kompletten Assembler, einen Disassembler und einen Einzelschritt-Simulator, der besonders für den Anfänger sehr nützlich ist. Natürlich zugeschnitten auf Ihren Computer, den COMMODORE 64. DAS MASCHINENSPRACHEBUCH ZUM COMMODORE 64, 1984, ca. 200 Seiten, DM 39,–.

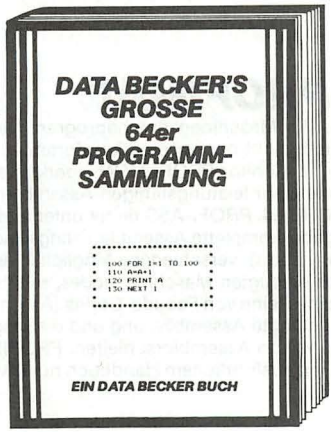
SIMON's BASIC ist ein Hit – wenn man es richtig nutzen kann. Deshalb gibt es jetzt zu dieser vielseitigen Befehlserweiterung unser umfangreiches Trainingsbuch, das Ihnen detailliert den Umgang mit den über 100 Befehlen des SIMON's BASIC erklärt. Ausführliche Darstellung aller Befehle (auch der, die nicht im Handbuch stehen!) Natürlich auch mit allen Macken und Hinweisen, wie man diese umgeht. Dazu zahlreiche Beispielprogramme und interessante Programmiertricks. Nach jedem Kapitel Testaufgaben zum optimalen Selbststudium. Dieses Buch sollte jeder SIMON's BASIC Anwender unbedingt haben! Das TRAININGSBUCH ZUM SIMON's BASIC, 1984, ca. 300 Seiten, DM 49,–.

Darauf haben Sie gewartet: Endlich ein Buch, das Ihnen ausführlich und verständlich die Arbeit mit der Floppy VC-1541 erklärt. DAS GROSSE FLOPPY BUCH ist für Anfänger, Fortgeschrittene und Profis gleichermaßen interessant. Sein Inhalt reicht von der Programmspeicherung bis zum DOS-Zugriff, von der sequentiellen Datenspeicherung bis zum Direktzugriff, von der technischen Beschreibung bis zum ausführlich dokumentierten DOS Listing, von den Systembefehlen bis zur detaillierten Beschreibung der Programme der Test/Demodiskette. Exakt beschriebene Beispiel- und Hilfsprogramme ergänzen dieses neue Superbuch. Mit dem GROSSEN FLOPPY-BUCH meistern Sie auch Ihre Floppy. DAS GROSSE FLOPPY BUCH, 1983, ca. 320 Seiten, DM 49,–.

DATA BECKER BÜCHER



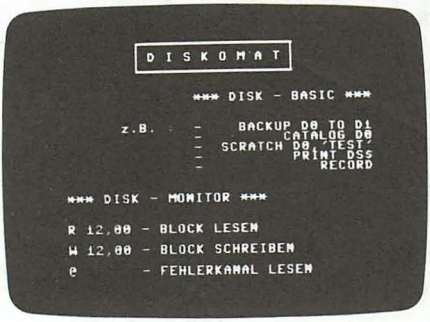
Der COMMODORE 64 ist ein Musikgenie, und mit diesem Buch lernen Sie alles über seine musikalischen Fähigkeiten. Der Inhalt reicht von einer Einführung in die Computermusik über die Erklärung der Hardware-Grundlagen und die Programmierung in BASIC bis hin zur fortgeschrittenen Musikprogrammierung. Zahlreiche Beispielprogramme und leicht verständliche Darstellung. Geschrieben vom Autor der bekannten Musikprogramme SYNTHIMAT und SYNTHESOUND.
Erschließen Sie sich die Welt des Sounds und der Computermusik mit dem MUSIKBUCH ZUM COMMODORE 64.
Ca. 200 Seiten, DM 39,–.



Diese neue, umfangreiche Programmsammlung hat es in sich. Über 50 Spitzenprogramme für den COMMODORE 64 aus den unterschiedlichsten Bereichen, vom Superspiel („Senso", „Pengo") über Graphik- und Soundprogramme (zum Beispiel „Fourier 64" oder „Orgel") sowie Utilities („Sort") bis hin zu Anwendungsprogrammen wie „Videothek" oder „Finanzbuchhaltung". Der Hit sind zu jedem Programm aktuelle Programmiertips und Tricks der einzelnen Autoren zum Selbermachen. Also – nicht nur abtippen, sondern auch dabei lernen und wichtige Anregungen für die eigene Programmierung sammeln.
Ca. 250 Seiten, DM 49,–.

Achtung Hobbyelektroniker: Diese Buch enthält nicht nur alles über Interfaces und Ausbaumöglichkeiten des COMMODORE 64, sondern auch über seine vielfältigen Einsatzmöglichkeiten von der Lichtorgel über Motorsteuerung, Spannungs- und Temperaturmessung bis zur programmierbaren Stromversorgung, und wie man diese verwirklicht. Zehn komplette Schaltungen zum Selberbauen, vom Eprommer über Eprom-Karte, Logic Analyzer, Freuqenzzähler, Hardware-Tracer, Pulsmeßgerät, Klatschschalter und Digital-Voltmeter bis zur preiswerten Spracheingabe-Sprachausgabe. Jeweils komplett mit Schaltplan, Layout und Softwarelisting.
Ca. 220 Seiten, DM 49,–, ab April 84.

Die neuen
DATA BECKER PROGRAMME



DISKOMAT

Dieses neue Spitzenpaket hilft Ihnen, mehr aus Ihrer Floppy zu machen, mit SUPERTWIN, dem Steuerprogramm, das zwei VC-1541 wie ein Doppellaufwerk verwaltet, mit DISK-BASIC, den Diskettenbefehlen des BASIC 4.0, mit denen Sie eine komplette Diskette oder Auszüge mit einem Befehl kopieren können und mit einem komfortablen DISK-MONITOR, mit dem Sie den Aufbau der Diskette erforschen und manipulieren können. Alles zusammen für nur DM 99,-.

PROFIMAT

Ein Spitzenpaket für Maschinenspracheprogrammierer. PROFIMAT enthält nicht nur unseren komfortablen Maschinensprache Monitor PROFI-MON, sondern auch PROFI-ASS, einen sehr leistungsfähigen Assembler für den COMMODORE 64. PROFI-ASS bietet unter anderem formatfreie Eingabe, komplette Assemblerlistings, ladbare Symboltabellen (Labels), verschiedene Möglichkeiten zur Speicherung des erzeugten Maschinencodes, redefinierbare Symbole, eine Reihe von Pseudo-Codes (Assembleranweisungen), bedingte Assemblierung und die Möglichkeit zur Erzeugung von Assemblerschleifen. PROFIMAT kostet komplett mit ausführlichem Handbuch nur DM 99,-.

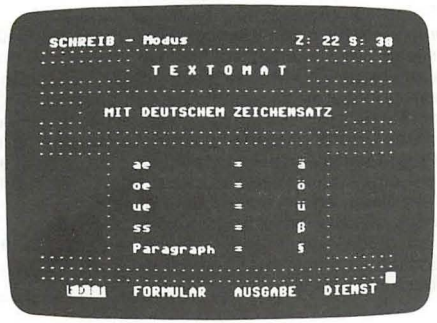
PASCAL 64

Jetzt können Sie die beliebte Sprache PASCAL auch auf dem COMMODORE 64 einsetzen. PASCAL 64 ist ein leistungsfähiger PASCAL-Compiler, der nicht nur den Befehlssatz des Standard PASCAL unterstützt, sondern auch die hochauflösende Graphik und die Sprites des Commodore 64. Ein-/Ausgabe über Diskette und Drucker sowie REAL und INTEGER Arithmetik. Unterprogramme aus Ihrer eigenen Programmbibliothek können vor dem Compilieren in Ihr Hauptprogramm mit eingebunden werden. PASCAL 64 ist sehr schnell, da echter Maschinencode erzeugt wird. PASCAL 64 kostet komplett mit ausführlichem Handbuch nur DM 99,-.

Die neuen
DATA BECKER PROGRAMME

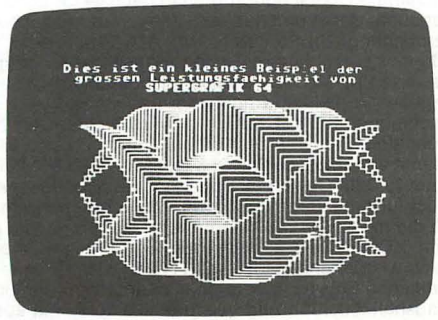
DATAMAT

Eine universelle Dateiverwaltung, die Sie von der Adressverwaltung über Mitgliederverwaltung bis hin zur Lagerbuchführung auf vielfältigste Weise nutzen können. Die frei gestaltbare Eingabemaske kann bis zu 50 Felder, max. 40 Zeichen pro Feld und bis zu 253 Zeichen pro Datensatz enthalten. Bis zu 2000 Datensätze pro Diskette sind möglich. Nach allen Feldern kann selektiert und sortiert werden, sogar nach mehreren gleichzeitig. Auswertungen können als Listen gedruckt oder in eine Datei als Verbindung zu TEXTOMAT geschrieben werden. DATAMAT ist (natürlich) menuegesteuert, in deutsch und dadurch extrem bedienerfreundlich. Ein Superprogramm, das zu jedem 64er gehören sollte. Komplett mit umfangreichem deutschen Handuch nur DM 99,-.



TEXTOMAT

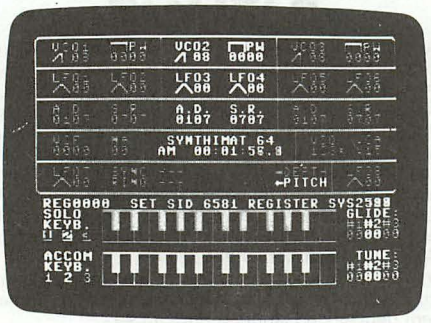
Ein außergewöhnliches Textverarbeitungsprogramm: 80 Zeichen pro Zeile durch horizontales Scrolling, Ausdruck bis zu 255 Zeichen, Textlänge bis zu 24000 Zeichen im Speicher, Verketten von Texten, umfangreiche Textbausteinverarbeitung und Formatierungsmöglichkeit, Formularsteuerung, Anpassung an fast jeden Drucker, Diskettenverwaltung, umfangreicher Befehlssatz und ca. 30 Steuerzeichen, Schnittstelle zu DATAMAT zur Erstellung von Rundschreiben mit individueller Anrede. TEXTOMAT ist komplett in Assembler geschrieben und dadurch extrem schnell. Menuesteuerung, deutsche Benutzerführung, natürlich deutscher Zeichensatz auf dem Schirm und ausführliches, 75seitiges Handbuch machen gerade für den Anfänger die Arbeit mit TEXTOMAT zum Kinderspiel und das zum sagenhaften Preis von nur DM 99,-.



SUPERGRAPHIK 64

Die neueste Version unserer beliebten SUPERGRAPHIK enthält jetzt über 30(!) Befehle zur Ausnutzung der fantastischen Möglichkeiten, die der 64 mit hochauflösender Graphik und Farbe bietet. Mit SUPERGRAPHIK 64 können Sie Punkte, Linien und Kreise ziehen. SPRITES definieren und manipulieren, Farben setzen, komplette Graphikbildschirme auf Diskette abspeichern bzw. laden und vieles andere mehr. Ergänzt wurde die SUPERGRAPHIK 64 zusätzlich um SUPERSOUND, eine neue Befehlserweiterung zur Nutzung der hervorragenden Soundmöglichkeiten des 64 und der Farb-Hardcopy auf dem neuen SEIKO GP 700 A. Mit SUPERGRAPHIK 64 machen Sie mehr aus Ihrem 64er, und das für nur DM 99,-.

Die neuen
DATA BECKER PROGRAMME

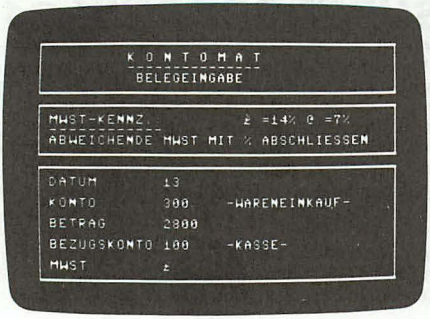


SYNTHIMAT

Mit diesem Superprogramm verwandeln Sie Ihren 64er in einen professionellen, polyphonen, dreistimmigen Synthesizer, mit dem Sie über die Tastatur ganze Akkorde spielen können. Zu den unglaublich vielen Möglichkeiten dieses Programms gehört auch die Bandaufnahme/-wiedergabe direkt auf bzw. von Diskette. SYNTHIMAT stellt gleichzeitig den Synthesizer optisch dar. Sämtliche Module sind farblich gekennzeichnet und übersichtlich angeordnet. Es ist ein Leichtes, mit SYNTHIMAT sämtliche Klangeigenschaften verschiedener Musikinstrumente zu imitieren, aber auch völlig neue Klangkreationen zu schaffen, selbst Weltraumklänge. Verwandeln Sie Ihren 64er für wenig Geld in eine Super-Musikmaschine mit SYNTHIMAT. Komplett mit ausführlichem Handbuch nur DM 99,-.

FAKTUMAT

Eine Sofortfakturierung mit integrierter Lagerbuchführung. Ideal für jeden Kleinbetrieb durch individuelle Anpassung von Steuersätzen, Maßeinheiten und Firmendaten an eigene Bedürfnisse. Natürlich sind auch die Kunden- und die Artikelstammdatei voll pflegbar. So können Sie beliebig den Umfang der Dateien wählen und diese Ihren Erfordernissen anpassen. Durch eine besondere Programmierweise ist es möglich, sehr schnell auf Kunden- und Artikeldaten zurückzugreifen. Der Zugriff auf diese Daten erfolgt jeweils über einen 6stelligen Schlüssel, den Sie frei definieren können. Die Fortschreibung von Artikel- und Kundendaten erfolgt selbstverständlich automatisch. Komplett mit ausführlichem Handbuch nur DM 99,-.



KONTOMAT

Ein Einnahme-Überschußprogramm nach §4 (3) EStG mit Kassenbuch, Bankkontenüberwachung, automatischer Steuerbuchung (Brutto und Netto), AfA Tabellenerstellung, Kontenblättern, Ermittlung der Ust.-Voranmeldungswerte und Monats- und Jahresabrechnung, geschrieben von einem Buchhalter und einem Programmierer. KONTOMAT ist voll parameterisiert (Firmendaten, Steuersätze, Kontennamen . . .) und läßt sich damit an Ihre Bedürfnisse anpassen. KONTOMAT ist geeignet für alle Gewerbetreibenden, die nicht laut HGB zur Buchführung verpflichtet sind. Mehrere Finanzämter haben die mit KONTOMAT ermittelten Daten bereits anerkannt. KONTOMAT ist menügesteuert und dokumentiert sich weitgehend selbst. KONTOMAT ist für den gewerblichen Einsatz aber auch als Lernprogramm oder zur Haushaltsbuchführung geeignet. Komplett mit ausführlichem Handbuch nur DM 99,-.